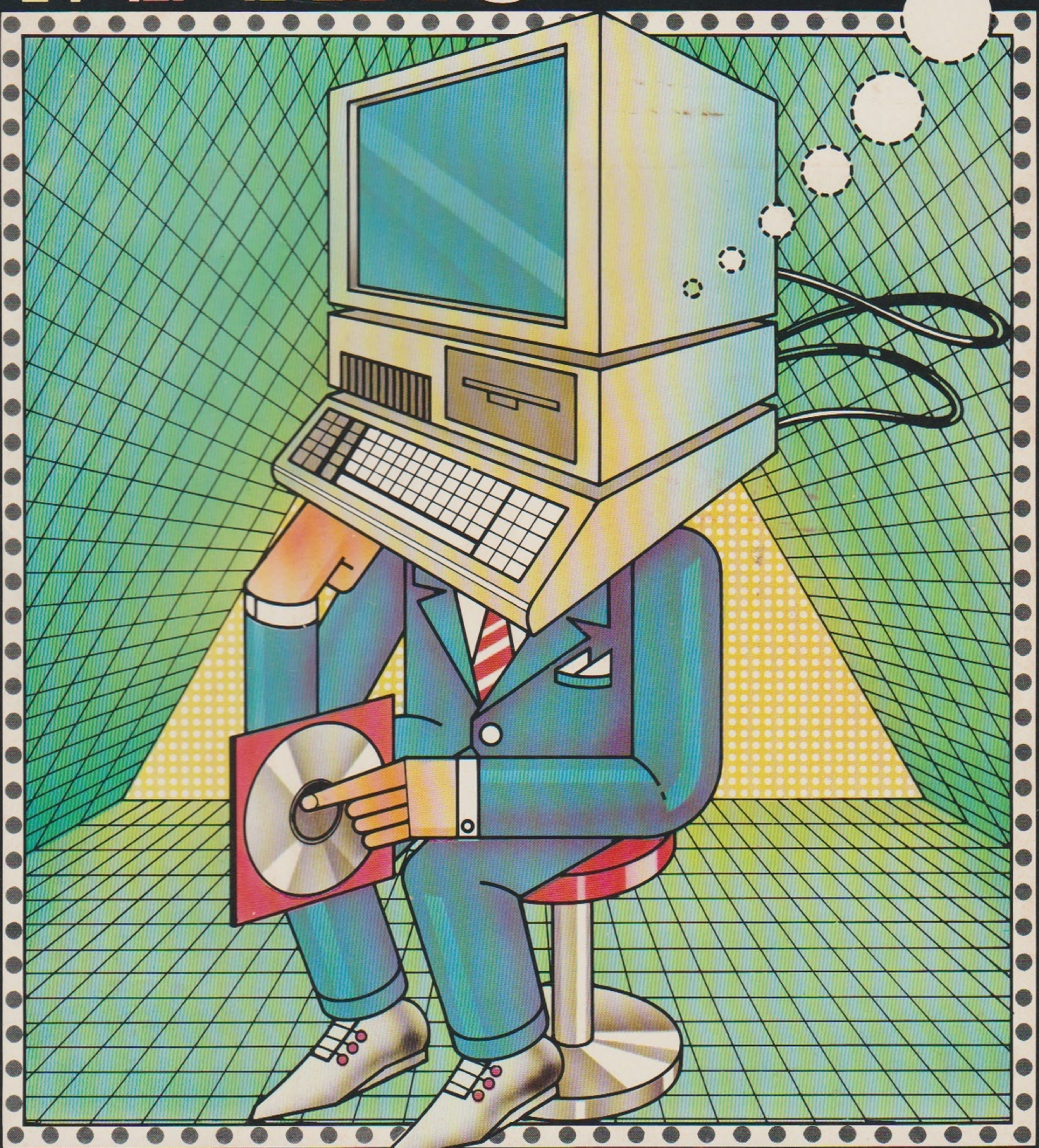
COMPUTER THINKING

NEWTECHBOOKS



0-590-33055-1

COMPUTER THINKING

SCHOLASTIC NEWTECHBOOKS

COMPUTER THINKING

NEWTECHBOOKS

by Patricia A. Relf

SCHOLASTIC INC.
New York Toronto London Auckland Sydney Tokyo

Staff for the *Scholastic New Tech Books*
Editorial Director: Eleanor Angeles
Cover Art Director: Bob Feldgus
Cover Illustration: Izumi Inoue

Computer Thinking

Developed and Produced for Scholastic Book Services® by MM Associates
Text Design and Illustrations: Richard Stalzer Associates, Ltd.

No part of this publication may be reproduced in whole or in part, or stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission of the publisher. For information regarding permission, write to Scholastic Inc., 730 Broadway, New York, NY 10003.

ISBN 0-590-33055-1

Copyright © 1984 by Scholastic Inc. All rights reserved. Published by Scholastic Inc.

12 11 10 9 8 7 6 5 4 3 4 5 6 7 8/8

Printed in the U.S.A. 14

CONTENTS

INTRODUCTION 4
Test Your Computer I.Q. 5

1 COMPUTERS 6
What Is a Computer? 6
How Do People Think? 8
How Do Computers Think? 10
What Is a Process? 11
Computer Anatomy 12
What Is a Program? 14
Running a Program 15
How Smart Is a Computer? 16
Review: COMPUTERS 18

2 FLOWCHARTS 19
Breaking Down a Task into Simple Steps 19
What Is a Flowchart? 20
Flowchart Symbols 22
Using Flowchart Symbols 24
Loops 25
Dead Ends 26
Review: FLOWCHARTS 28

3 PROGRAMS 30
Flowcharts and Programs 30
Programs and Data 31
Sequence 32
Numbering Lines in a Program 34
Running a Program in Order 36
Making Changes 38
Debugging 40
Branching 42
Branching Programs 43
IF . . . THEN Statements 44
Logic: *And* and *Or* 45
Logic: A Quick Quiz 46
GO TO . . . 47
GOTO 48
Loops 49
Endless Loops 50
Debugging: Branches and Loops 52
Review: PROGRAMS 54

4 CODES 56
In Other Words 56
Symbols and Codes 58
Variables 59
Strings 60
PRINT 61
Making More Storage Space 62
Arrays 64
Using Arrays 65
Using Variables in Programs 66
Review: CODES 68

5 ACCESS TO COMPUTERS 69
Commands and Statements 69
Writing Programs 70
Commands 71
Graphics 72
Uses of Computers 74
Review: ACCESS TO COMPUTERS 76

GLOSSARY 77

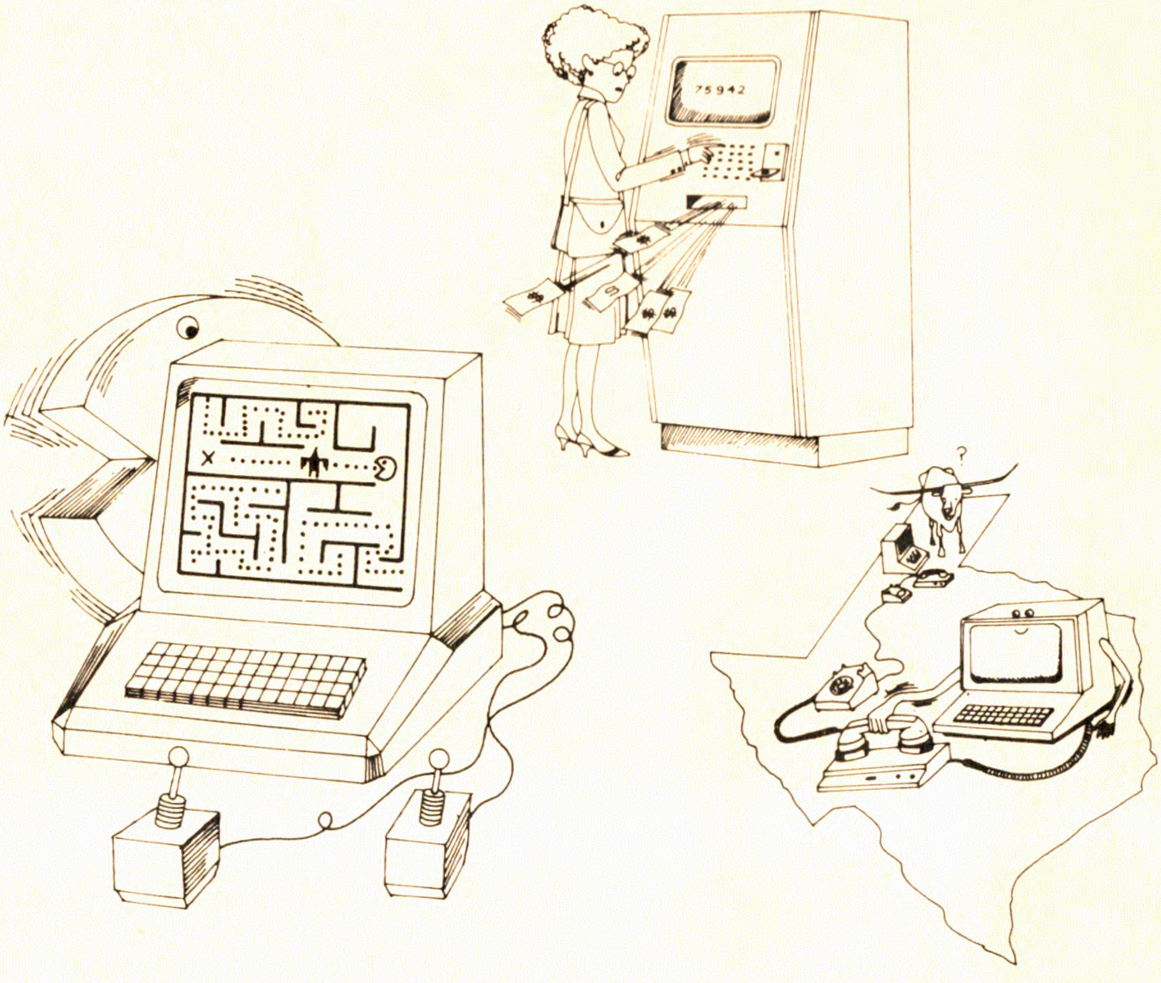
ANSWER KEY 79

INTRODUCTION

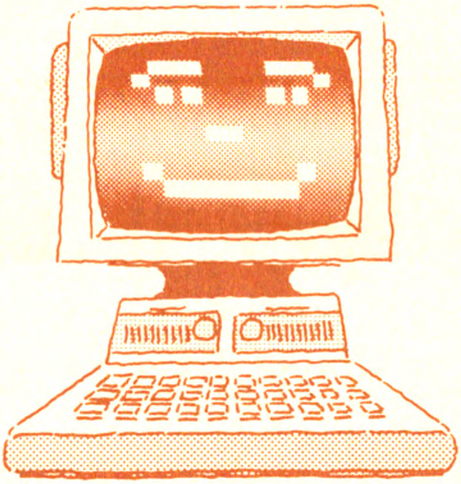
If you've ever played a video game, made a long-distance phone call, or deposited money at a bank, you've put a computer to work. These days, computers do all kinds of jobs, but they're just useless lumps of machinery without *people* to tell them exactly what to do and how to do it. That's the job of a computer programmer.

Completing this book won't turn you into an expert programmer, but you will learn how to *think* like a programmer. You'll sharpen your logic skills. You'll learn what a computer can do. And you'll learn how to "talk" to a home or school computer and to give it some simple commands.

Start by taking the quiz on the next page. You may already know more about computers than you think!



Test Your Computer I.Q.



How much do you know about computers?

Next to each statement, write **T** *for* true *or* **F** *for* false.

____ **1.** A computer can do anything a human brain can do.

____ **2.** Different computers "speak" different languages.

____ **3.** Computers can only do what people tell them to.

____ **4.** A computer has a *memory*.

____ **5.** Only computer programmers can use computers.

____ **6.** When the person using a computer makes a mistake, the computer gets angry.

____ **7.** A set of instructions that a computer can follow is called a *program*.

____ **8.** Computers can use disks or cassette tapes to store information.

____ **9.** A *flowchart* is a diagram of a computer's electrical circuits.

____ **10.** Computers will take over the world.

1 COMPUTERS

What Is a Computer?

A computer is a machine. Like any other machine, it's a tool that people have built to make work easier. A computer is not smart; people have to tell it exactly what to do, step by step.

A computer can

- store a huge amount of information in its memory and recall it when needed.
- do calculations quickly.
- sort things.
- do repetitious jobs.
- combine facts that people give it.

A computer cannot

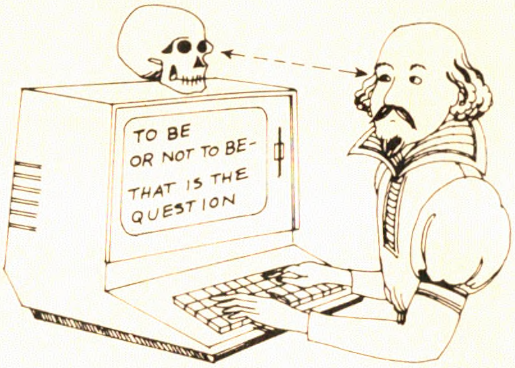
- feel emotions such as happiness, sadness, or anger.
- make moral judgments about what's right and wrong.
- think up its own new ideas.

Some jobs make good use of the things computers can do. Other jobs can be done better by people without a computer's help.

Carefully reread the list of things that computers can and cannot do. Then look at each job. If a computer could do the job faster or better than a person, write computer *in the blank. If a person could do the job better without a computer, write* person.

Job

1. Check the division problems from your math homework ____________________
2. Write a poem ____________________


TO BE OR NOT TO BE- THAT IS THE QUESTION

3. Keep a list of your 50 favorite elephant jokes

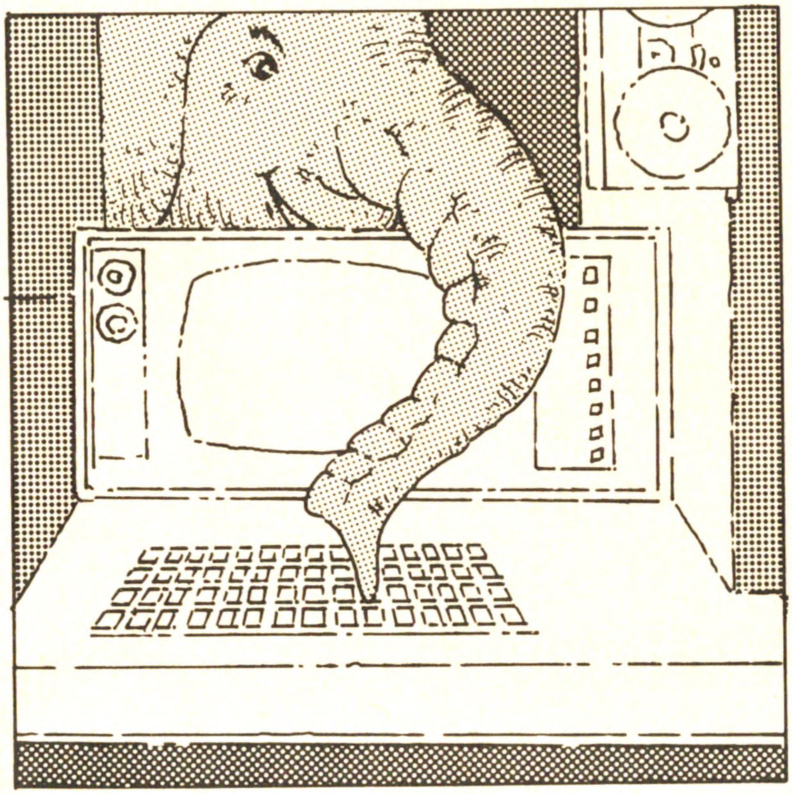
4. Alphabetize the titles of 30 books about armadillos

5. Decide whether stealing is right or wrong

6. Use past weather statistics to forecast tomorrow's weather

Name another job a computer could do well.

Name another job a computer couldn't do.

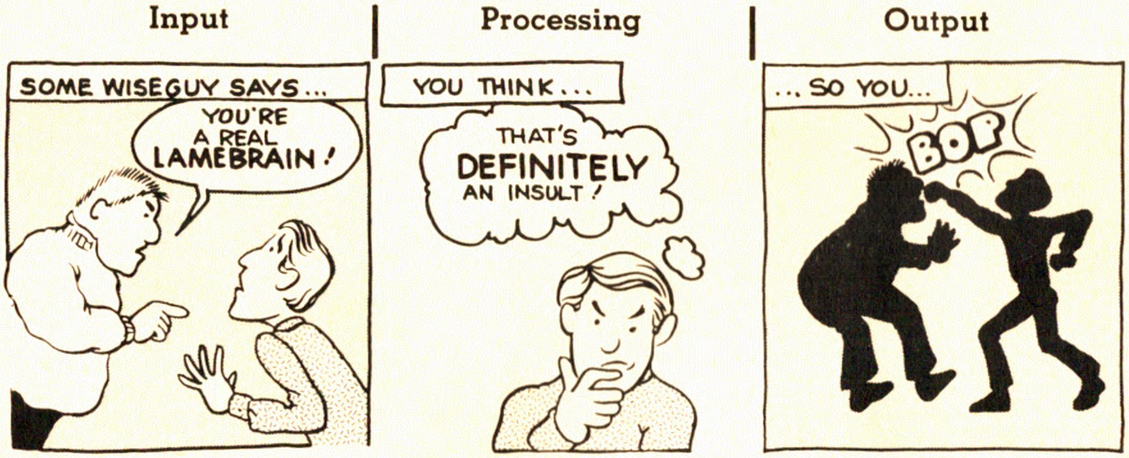


How Do People Think?

Did you ever think about how you think? Information goes *into* your brain through your senses (sight, hearing, taste, touch, and smell), your brain works on that information, and you then respond.

- The information that goes in is **input**.
- Your brain **processes** the information.
- What you do as a result is **output**.

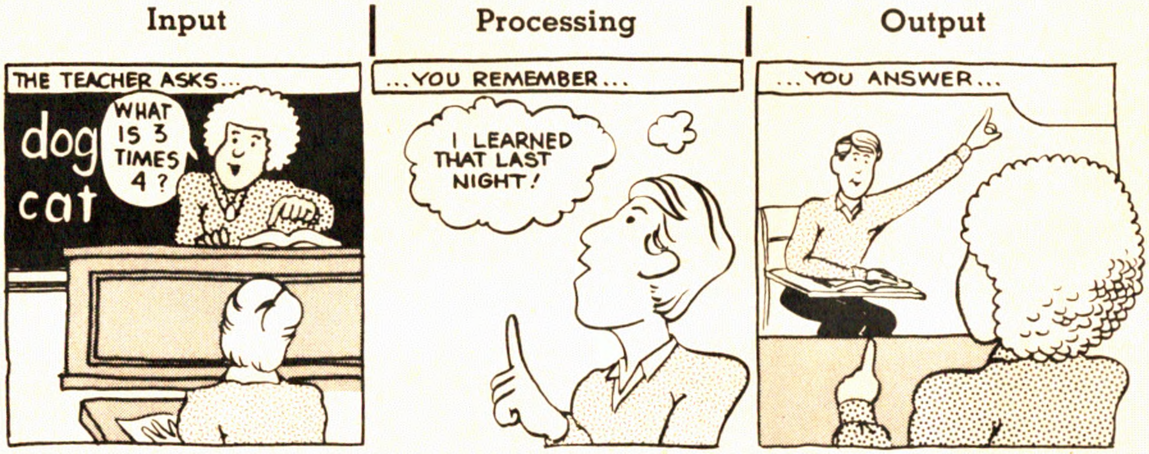
Here's an example:


Input
Processing
Output
SOME WISEGUY SAYS...
YOU'RE A REAL LAMEBRAIN!
YOU THINK...
THAT'S DEFINITELY AN INSULT!
... SO YOU...
BOP

Here are more examples of input, processing, and output.

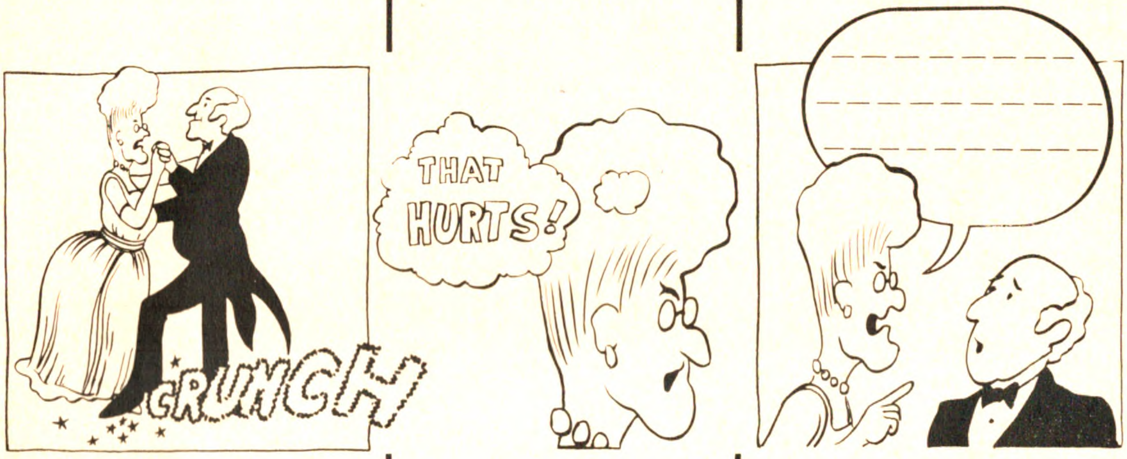
Fill in the blanks with a suitable answer.

1

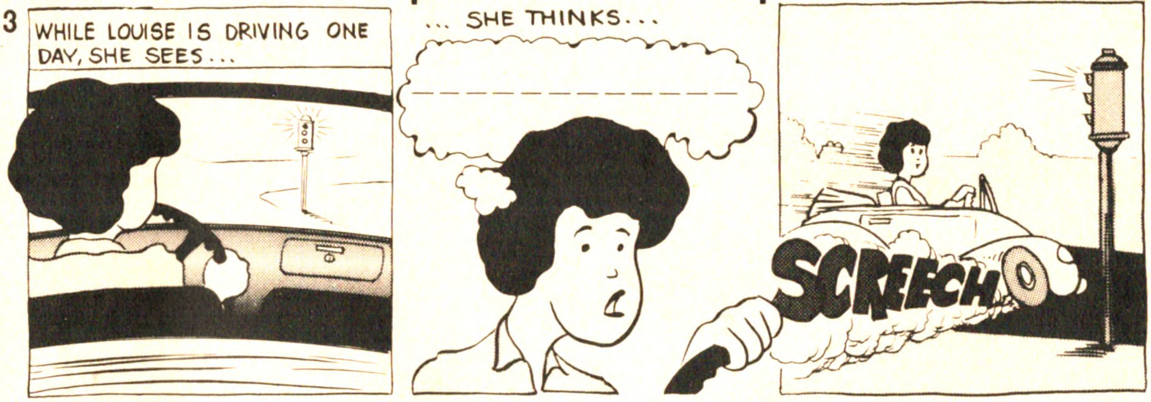

Input
Processing
Output
THE TEACHER ASKS...
dog
cat
WHAT IS 3 TIMES 4?
...YOU REMEMBER...
I LEARNED THAT LAST NIGHT!
...YOU ANSWER...

Input | **Processing** | **Output**

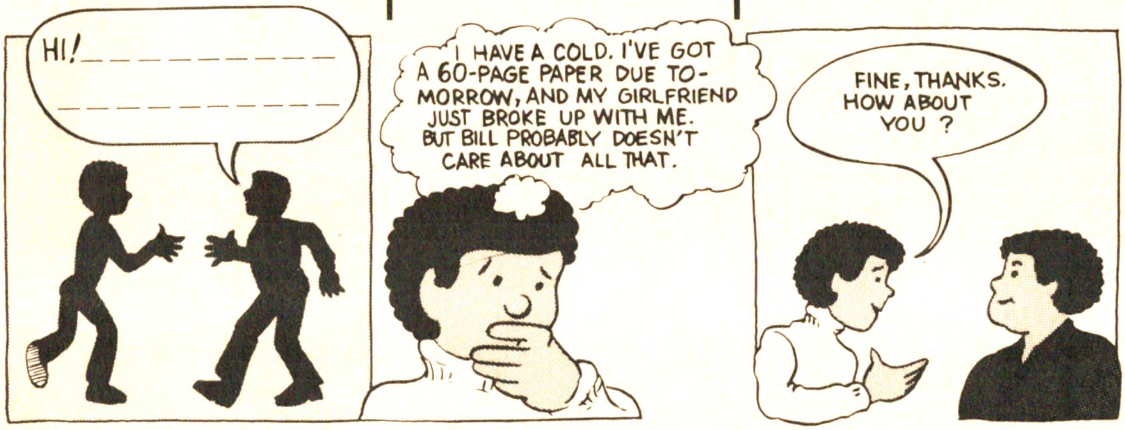
2


CRUNCH
THAT HURTS!

3


WHILE LOUISE IS DRIVING ONE DAY, SHE SEES...
... SHE THINKS...
SCREECH

4


HI!
I HAVE A COLD. I'VE GOT A 60-PAGE PAPER DUE TOMORROW, AND MY GIRLFRIEND JUST BROKE UP WITH ME. BUT BILL PROBABLY DOESN'T CARE ABOUT ALL THAT.
FINE, THANKS. HOW ABOUT YOU ?

How Do Computers Think?

In some ways, a computer works like your brain. Suppose you gave a computer a mixed-up list of names and asked it to put them into alphabetical order.

- The mixed-up list is the **input**.
- The computer sorts the names into alphabetical order; it **processes** them.
- The alphabetized list that the computer prints on a TV screen or on paper is the **output**.

Here are four jobs that a computer could do. For each job, what input *would you give the computer? What* output *would the computer give you? Write your answers in the blanks.*

Example: Subtract three from six.
Input: 6 – 3 *or* what is 6 – 3?
Output: 3

1. Add six and three.

Input: ____________________

Output: ____________________

2. Calculate the average of ten test scores.

Input: ____________________

Output: ____________________

3. Calculate how many apples are needed for a worm dinner party.

Input: (Hint: You will need to use two numbers) ____________________

Output: ____________________

4. Calculate how many days there are between today and your next birthday.

Input: ____________________

Output: ____________________

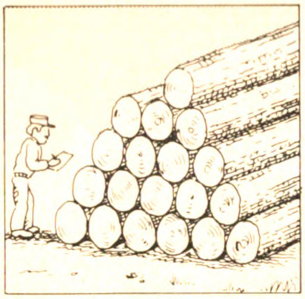
Something to think about: What **process** would the computer go through to do each job above? What would it need to know how to do?

____

What Is a Process?

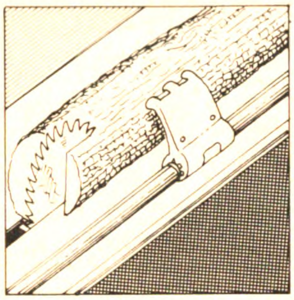
If you watched a lumber mill from a short distance away, you would see big, round logs going *into* the mill, and flat boards coming *out*. What happens in between? Inside the mill, logs are sawed into boards. Sawing is the **process** that turns logs into boards. A **process** is a series of operations, actions, or changes that turns raw material—**input**—into a final product—**output**.

Input



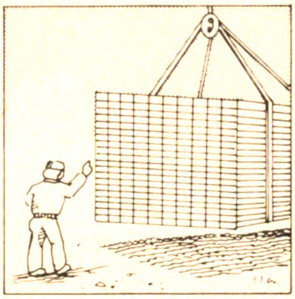
Logs

Process



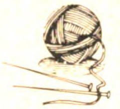
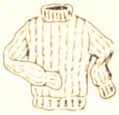
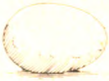
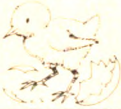
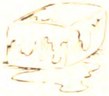
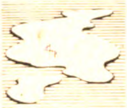
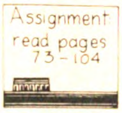
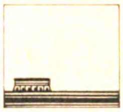
Sawing

Output

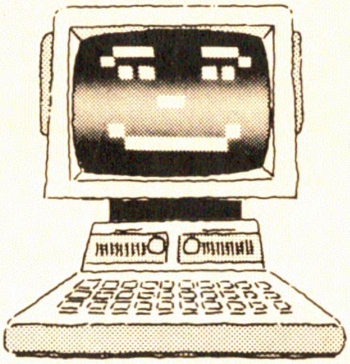


Boards

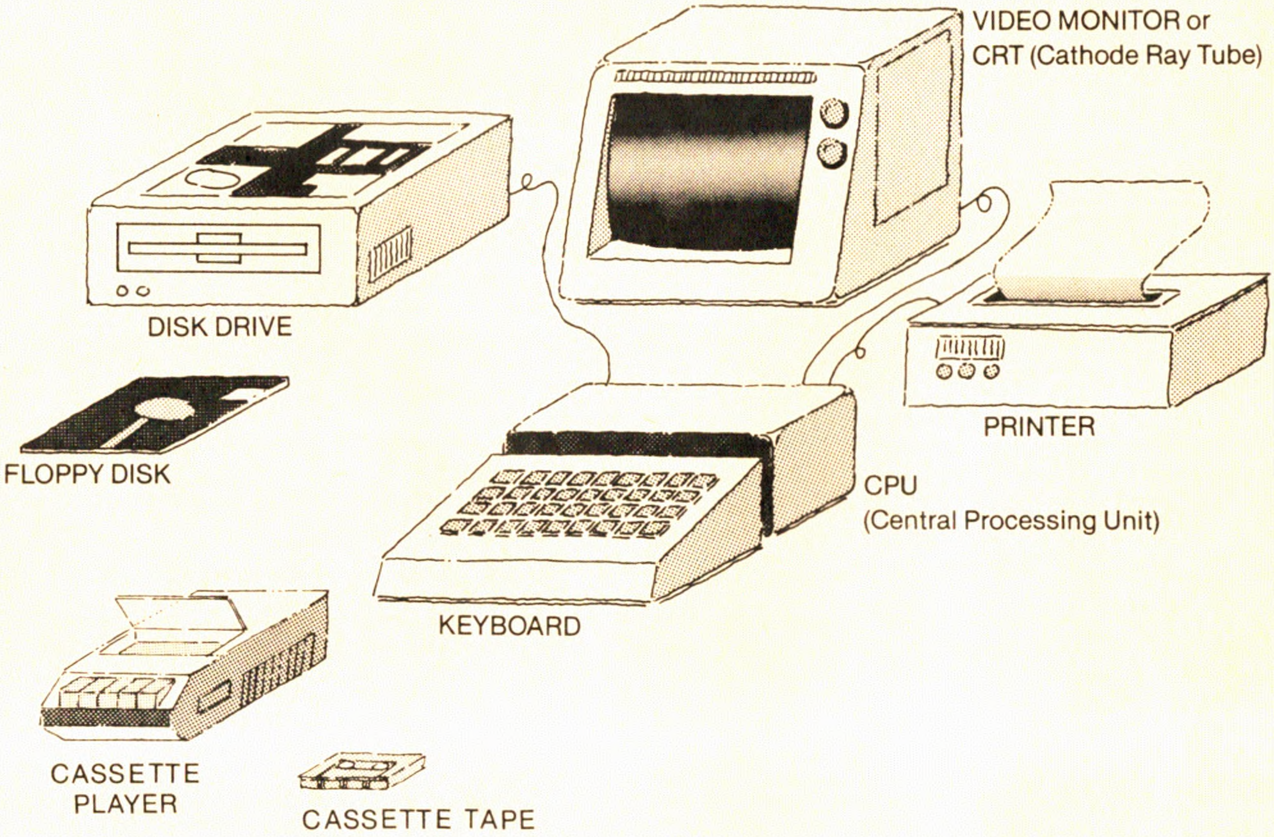
For each item below, write the name of the process *that turns the* input *in the left column into the* output *in the right column. The first example is done for you.*

	Input	Process	Output
1.		knitting	
2.			
3.			
4.			
5.	3, 2		5
6.	21, 7		3

Computer Anatomy



There are many different kinds of computers. Some are big; some are extremely small; some have many pieces; others are contained in one package. The picture below shows an imaginary computer. A real one at home or in school might have all or only some of these parts.


VIDEO MONITOR or
CRT (Cathode Ray Tube)
DISK DRIVE
FLOPPY DISK
PRINTER
CPU
(Central Processing Unit)
KEYBOARD
CASSETTE
PLAYER
CASSETTE TAPE

Read the following descriptions to find out what each part of the computer does. Then fill in the numbered blanks.

- *If the part helps the computer take in information, write* input.
- *If the part works on the information received, write* processing.
- *If the part gives the computer's results, write* output.

______________________ 1. **CPU (Central Processing Unit):** The main "thinking" part of the computer. This part does the computer's real work—calculating, sorting, comparing, and so on. Inside the CPU is a collection of tiny electrical circuits.

______________________ 2. **Keyboard:** The keyboard looks like a typewriter. It is used to type words, numbers, and symbols. It gives the computer information and instructions.

______________________ 3. **Video monitor or CRT (Cathode Ray Tube):** A screen that displays numbers, letters, and even pictures. It looks like a television screen. The video monitor is one way for the computer to give a message or an answer.

______________________ 4. **Printer:** Prints letters, numbers, and symbols on paper. This is another way for the computer to give a message or an answer.

______________________ 5. **Floppy disk:** A small, flat, flexible object made of magnetic material. Information and instructions for the computer can be recorded and stored on the disk and then put into the computer whenever needed.

______________________ 6. **Disk drive:** A machine that reads the instructions and information from a disk and sends them into the main part of the computer.

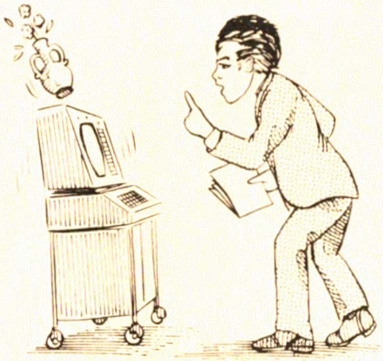
______________________ 7. **Cassette tape:** Used by some computers instead of a floppy disk. It does exactly the same thing—stores instructions and information to put into the computer.

______________________ 8. **Cassette player:** A machine that reads the instructions and information from a cassette tape and sends them into the main part of the computer. It takes the place of a disk drive for some computers.

What Is a Program?

A computer can't do even the simplest job unless a person tells it exactly what to do, step by step. A **program** is a set of instructions that tells a computer how to do a job. A program is made up of many individual instructions, or **program statements**. Each program statement has a **line number** so the computer will do the steps in the right order.

Here is a kind of program that people use every day:



1 Pour half a cup of milk into a blender.
2 Add 2 tablespoons of chocolate syrup.
3 Add 2 scoops of chocolate ice cream.
4 Cover the blender and turn it on for 30 seconds.
5 Pour the mixture into a glass.

1. What job does this program tell you how to do?

2. How many program statements does this program have?

3. What is the line number of the statement that tells you how much ice cream to use?

4. Which program statements would you have to change if you wanted a different flavor?

5. Write your own program statement to add to the end of this program. (Don't forget to give it a line number.)

Running a Program

When you tell a computer to **RUN** a program, the computer follows all the instructions in the program, statement by statement, starting with the lowest line number. In other words, the computer *does* what the program tells it to do.

You be the computer. Follow all the instructions below—that is, RUN this program. Then answer the questions that follow by writing the correct answers in the blanks.

```
1 Start with your age.
2 Add 10.
3 Subtract 3.
4 Add 14.
5 Subtract 6.
6 Subtract your age.
7 Add 2.
8 Print your answer in the space below.
```

1. How many statements does this program have? ______________________

2. Which statement tells you to give your output? ______________________

3. If someone 19-years-old ran this program, what would his or her output be?

____

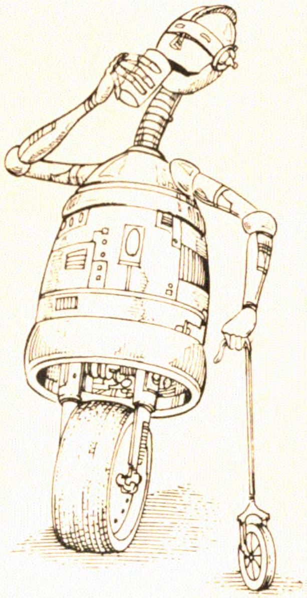
4. If the program were changed so that line 3 read "Add 4," what would the 19-year-old person's output be? ______________________

How Smart Is a Computer?

How smart is a computer? Not very! When you write a computer program, you have to tell the computer exactly how to do every little thing. It's like a very obedient but not very bright robot. It will do everything you tell it to do, but it won't do *anything* you *don't* tell it to do. So a computer won't correct your mistakes or fill in any instructions you leave out.

Imagine that you are telling a robot how to take a sip of water from a glass. It's not as easy as it sounds! The program might be something like this:

1 Hold the glass in your hand.
2 Make sure the open end is at the top!
3 Lift the glass until its edge is against your bottom lip.
4 Open your lips slightly.
5 Very slowly, tilt the glass towards you.
6 As soon as you feel some water in your mouth, stop tilting the glass.
7 Close your lips and start tilting the glass away from you.
8 When the glass is level, stop tilting it.
9 Swallow the water that's in your mouth.
10 Lower the glass back to the table.



In the blank, write the answer to each question. Remember that the robot—like a computer—will follow your instructions exactly!

1. What would happen if you left line 3 out of the program?

____

2. What would happen if you left out line 4?

____

3. Which line tells the robot to start pouring water into its mouth?

____

4. Which line would you change if you wanted the robot to take a big gulp of water instead of a sip?

____

5. Which lines would you have to change if you wanted the robot to drink from a cup instead of a glass?

____

A simple task like walking is also difficult to explain to someone who has never done it.

Number 1 below is the first line of a program that tells a robot how to take one step. In the blanks, write the next two lines of the program.

1 Lift your right foot.

2 ______________________________

3 ______________________________

Now write the next three lines of the program to tell the robot how to take the next step.

4 ______________________________

5 ______________________________

6 ______________________________

Something to think about: Could you add one more line to the program that would keep the robot walking continuously?

Review: COMPUTERS

Read each sentence below. From the list of words and phrases at the right, choose the one that correctly completes each sentence. Write the word(s) in the blank.

1. A ____________________ is a machine that helps people do such jobs as calculating, filing, and sorting.
2. The ____________________ is the main "thinking" part of the machine described in sentence 1 above.
3. Any series of operations, actions, or changes that turns raw materials into a final product is called a ____________________.
4. The raw material—information—given to a computer is called ____________________.
5. The ____________________ is used to type information into a computer.
6. The results that a computer gives out are called ____________________.
7. A computer can show its answers and messages on a screen called a ____________________ or on paper using a machine called a ____________________.
8. A ____________________ is a set of instructions that tells a computer how to do a job.
9. A single instruction is called a ____________________.
10. When a computer is told to ____________________ a program, it follows all the instructions, one by one, in order.

output

input

process

CPU (Central Processing Unit)

program

program statement

video monitor

printer

keyboard

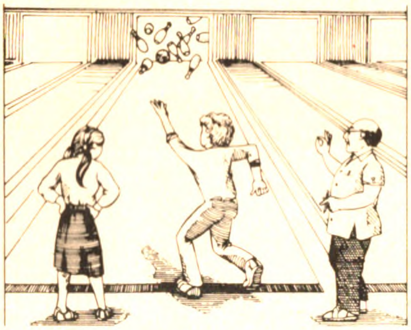
computer

RUN

2 FLOWCHARTS

Breaking Down a Task into Simple Steps

When you give instructions, whether to a person or to a computer, it's important to break the job down into simple steps. Imagine, for example, that you wanted to tell your friend how to bowl. What are the *main steps*? The main steps are:



1 Pick up bowling ball correctly.
2 Aim ball at pins.
3 Roll ball down alley.
4 Record score.

A complicated and difficult job often seems easier when you break it down into simple steps in this way. What are the main steps in each of the following jobs?

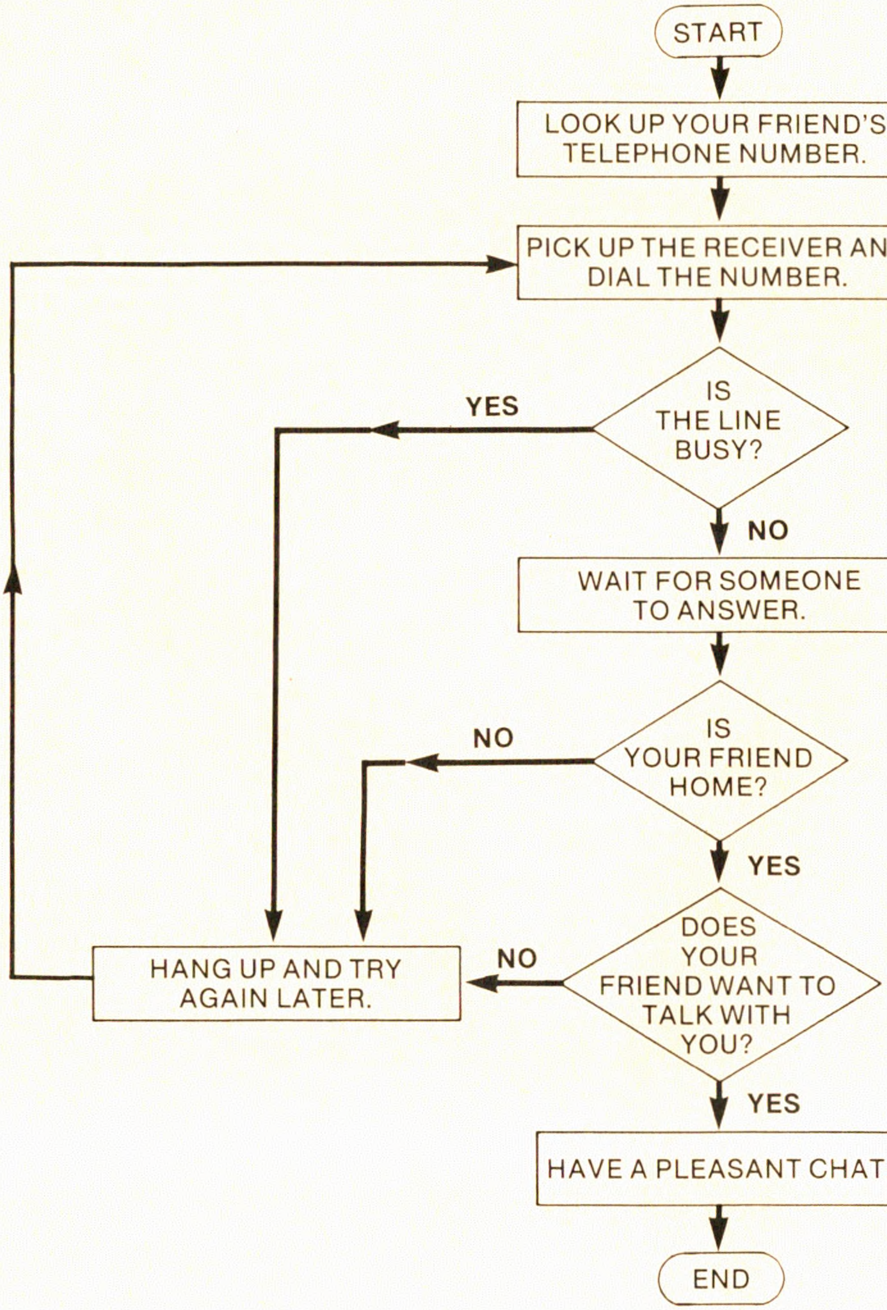
1. Growing a flower from a seed

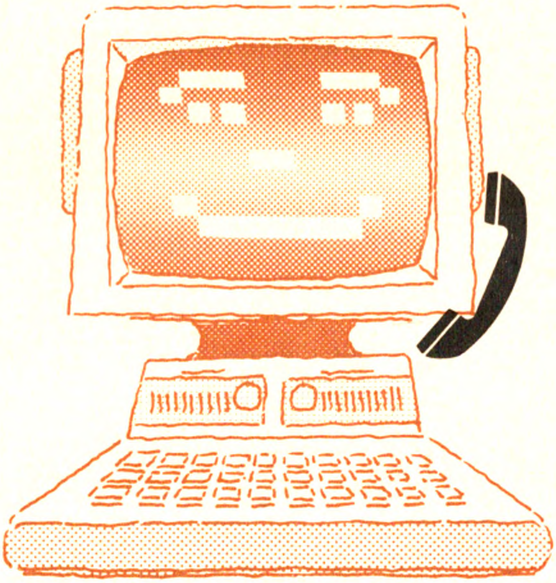
2. Changing a car's flat tire

3. Writing a book report

What Is a Flowchart?

A **flowchart** is a map of a task or a program that breaks a job down into simple steps. Arrows on a flowchart show which step comes next. This flowchart shows the steps involved in telephoning a friend.


START
LOOK UP YOUR FRIEND'S TELEPHONE NUMBER.
PICK UP THE RECEIVER AND DIAL THE NUMBER.
IS THE LINE BUSY?
YES
NO
WAIT FOR SOMEONE TO ANSWER.
IS YOUR FRIEND HOME?
NO
YES
DOES YOUR FRIEND WANT TO TALK WITH YOU?
NO
YES
HANG UP AND TRY AGAIN LATER.
HAVE A PLEASANT CHAT.
END



Use the flowchart on the opposite page to answer the questions below.

1. If you want to telephone a friend, what do you do first?

____

2. If you get a busy signal, what do you do?

____

3. If your friend's phone rings, what do you do?

____

4. What happens if your friend is angry with you and refuses to talk with you?

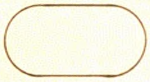
____

5. How many times does the flowchart offer you a *choice* of things to do next?

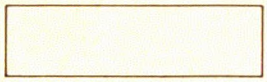
____

Flowchart Symbols

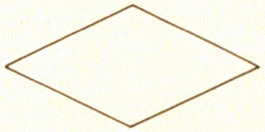
Each shape in a flowchart stands for a different kind of step in the program.



is the symbol for START and END. It shows where the program begins and stops.

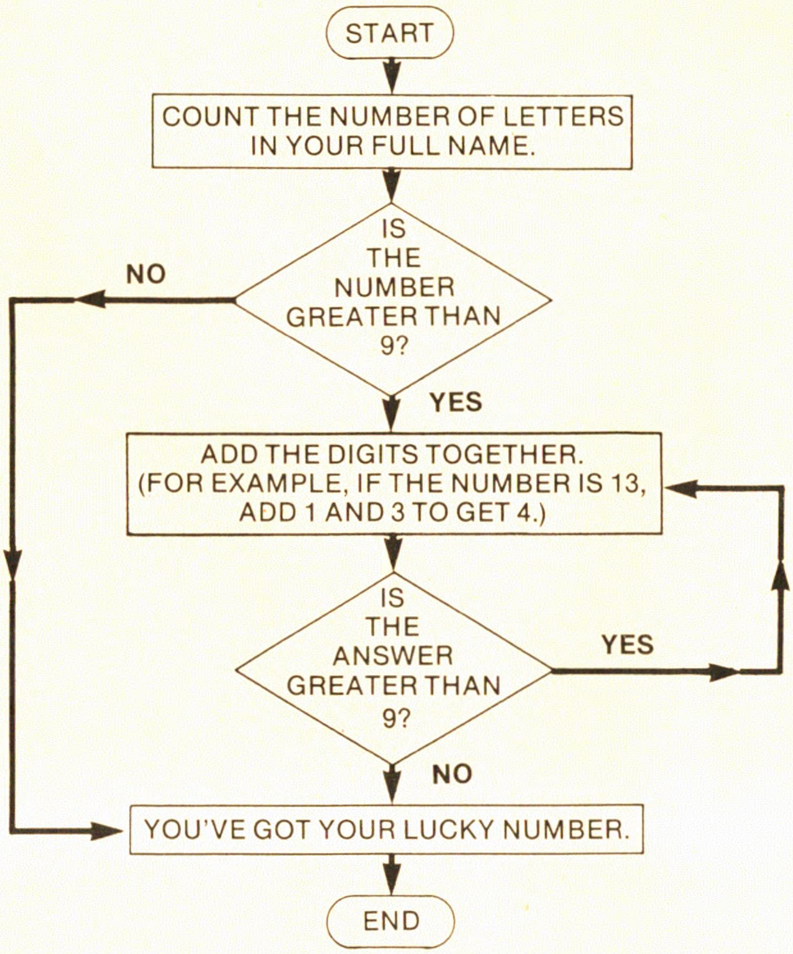


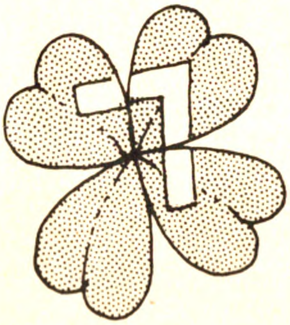
stands for an OPERATION. An operation tells what to do.



is the symbol for DECISION. It gives a choice. Inside the diamond is a question that determines what to do next.

This flowchart tells you how to figure out your "lucky number."


START
COUNT THE NUMBER OF LETTERS IN YOUR FULL NAME.
IS THE NUMBER GREATER THAN 9?
NO
YES
ADD THE DIGITS TOGETHER. (FOR EXAMPLE, IF THE NUMBER IS 13, ADD 1 AND 3 TO GET 4.)
IS THE ANSWER GREATER THAN 9?
YES
NO
YOU'VE GOT YOUR LUCKY NUMBER.
END



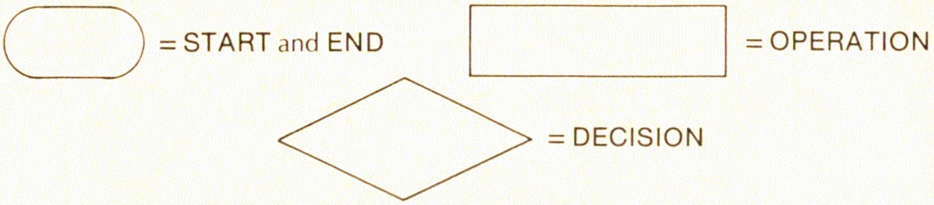
Use the flowchart on the opposite page to answer the questions below.

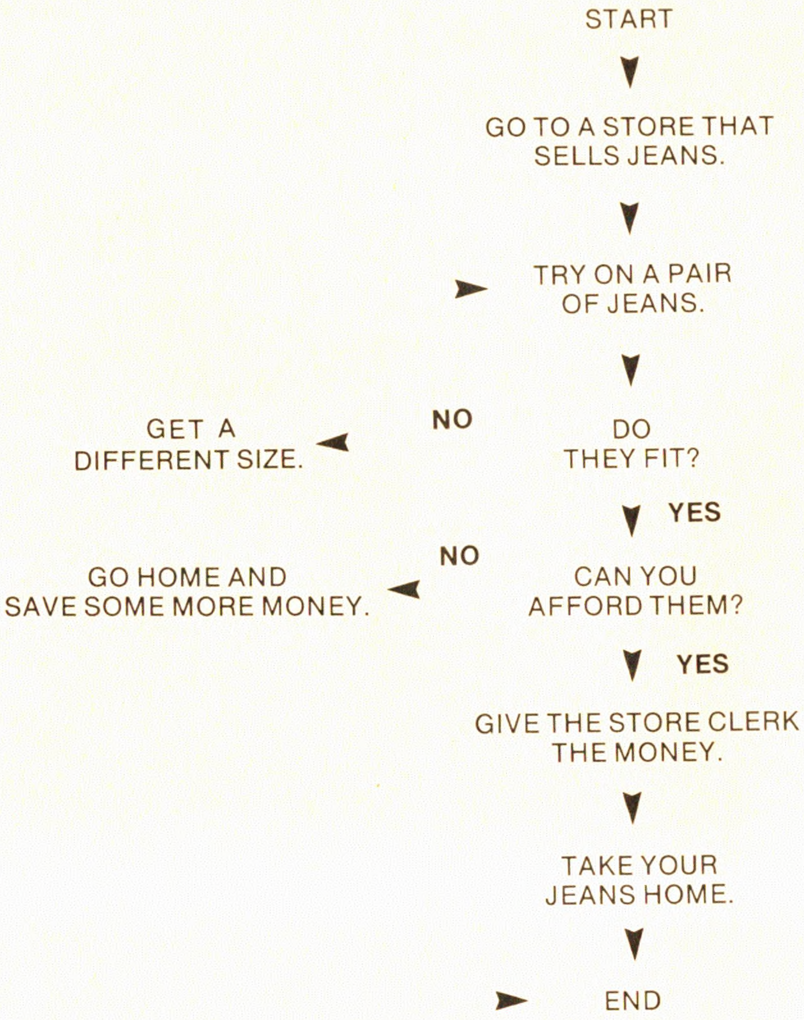
1. What is the very first step in the program? ______________________
2. How many *operation* steps are there in this flowchart? ______________
3. How many *decision* steps are there? ______________________
4. How many choices does the first decision step give you? ____________
5. What is your "lucky number"? ______________________

Using Flowchart Symbols

The flowchart below tells how to buy a pair of jeans. The words are there, but the symbols are missing.

Draw the correct shape around each step. Remember:

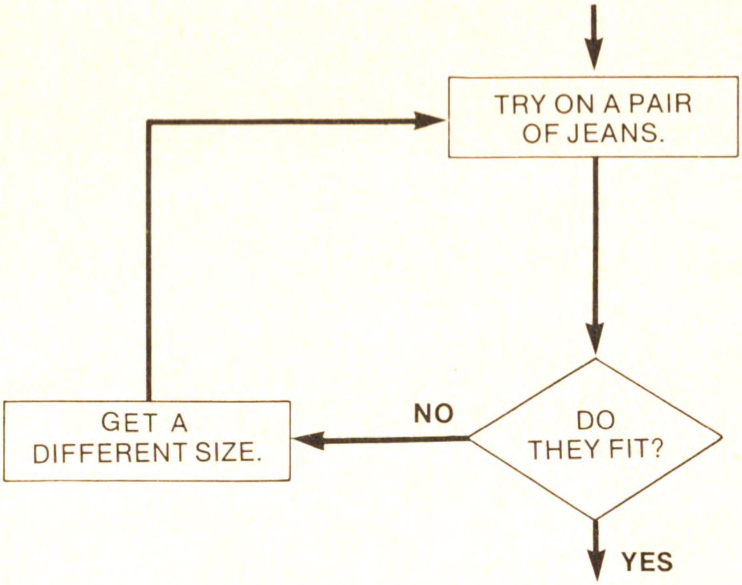

= START and END
= OPERATION
= DECISION


START
GO TO A STORE THAT SELLS JEANS.
TRY ON A PAIR OF JEANS.
DO THEY FIT?
NO
GET A DIFFERENT SIZE.
YES
CAN YOU AFFORD THEM?
NO
GO HOME AND SAVE SOME MORE MONEY.
YES
GIVE THE STORE CLERK THE MONEY.
TAKE YOUR JEANS HOME.
END

Loops

Flowcharts often have a step—or a section of several steps—that can be repeated over and over. A section that can be repeated is called a **loop**.

Here is a loop from the flowchart on page 24.


TRY ON A PAIR OF JEANS.
GET A DIFFERENT SIZE.
NO
DO THEY FIT?
YES

Use the loop above to answer the questions below.

1. How many times could you possibly repeat the loop?

2. When would you stop repeating the loop?

3. What steps are included in the loop?

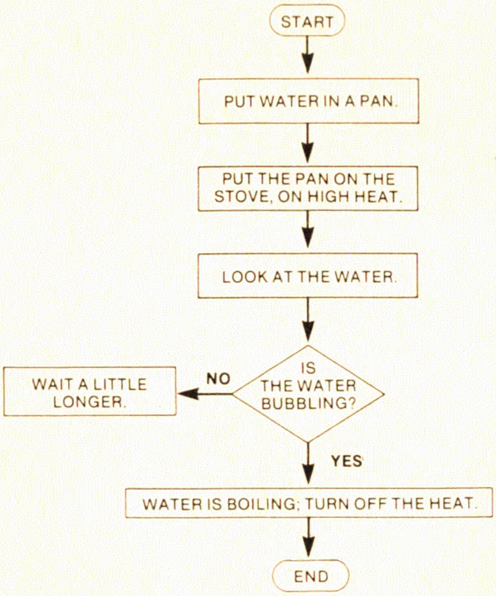
4. How would it be possible to complete the program without entering the loop?

Dead Ends

A flowchart may have many different paths, but every path must lead from the START to the END. Every step has an arrow pointing to the next step. A step without an arrow showing what to do next is called a **dead end**. (The END circle is the only step that *should* be a dead end.)

The flowchart below has a dead end. Draw in the missing arrow so the flowchart will no longer have a dead end.

How to Boil Water


START
PUT WATER IN A PAN.
PUT THE PAN ON THE STOVE, ON HIGH HEAT.
LOOK AT THE WATER.
IS THE WATER BUBBLING?
NO
WAIT A LITTLE LONGER.
YES
WATER IS BOILING; TURN OFF THE HEAT.
END

Use the flowchart above to answer the following questions.

1. What steps are included in the loop you made by drawing in the missing arrow?

____

____

____

2. When would you stop repeating the loop?

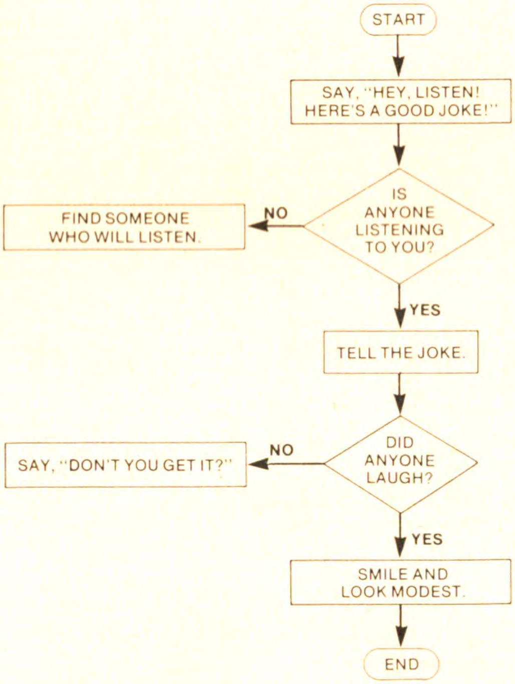
____

3. How would it be possible to complete the program without entering the loop?

____

Here's a flowchart showing how to tell a joke. The flowchart has two dead ends. Draw in the missing arrows so the flowchart will no longer have dead ends.

How to Tell a Joke


START
SAY, "HEY, LISTEN! HERE'S A GOOD JOKE!"
IS ANYONE LISTENING TO YOU?
NO
FIND SOMEONE WHO WILL LISTEN.
YES
TELL THE JOKE.
DID ANYONE LAUGH?
NO
SAY, "DON'T YOU GET IT?"
YES
SMILE AND LOOK MODEST.
END

Use the flowchart above to answer the following questions.

1. What steps are included in the first loop you made?

____

____

____

2. When would you stop repeating that loop?

____

3. How would it be possible to complete the program without entering the loop?

____

4. What steps are included in the second loop you made?

____

____

____

Review: FLOWCHARTS

Use the steps below to draw your own flowchart showing how to make popcorn. Use arrows to show what to do next. Include two loops. Don't leave any dead ends.

Put a little oil into a pan.

Put the pan on medium high heat.

Pour in popcorn and cover pan.

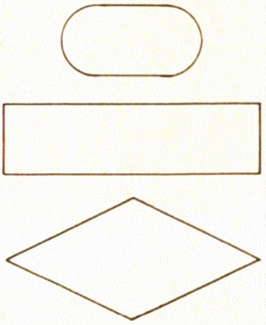
When you hear corn starting to pop,
shake the pan.

If you still hear corn popping,
shake the pan some more.

If the corn stops popping,
take the pan off the heat.

Take the lid off the pan and eat the popcorn.

Remember these symbols:



= START and END

= OPERATION

= DECISION

Draw your own flowchart below.

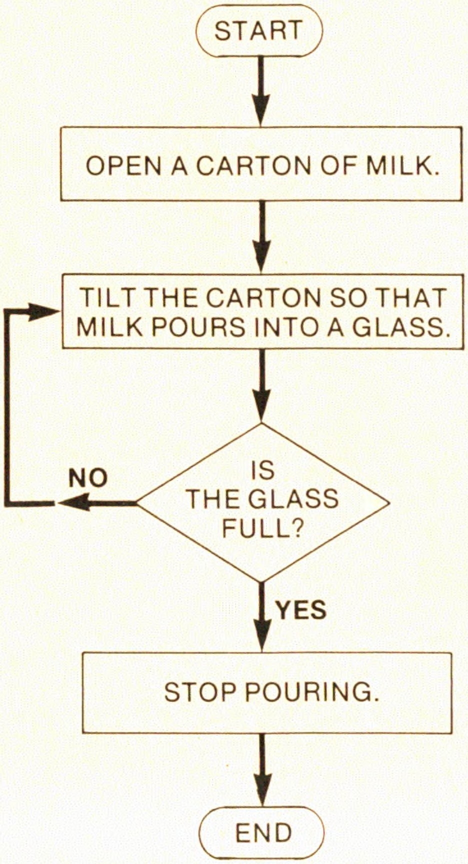
3 PROGRAMS

Flowcharts and Programs

What is the difference between a **flowchart** and a **program**?

A flowchart	**A program**
■ is a tool to help you organize *before* you write a program.	■ is a detailed set of instructions that a computer can understand.
■ shows the main steps in a job.	■ tells a computer exactly how to do a job.
■ uses symbols for different kinds of steps.	■ is a numbered list of instructions in computer language.

Look at the flowchart below. Then write the steps in the box.

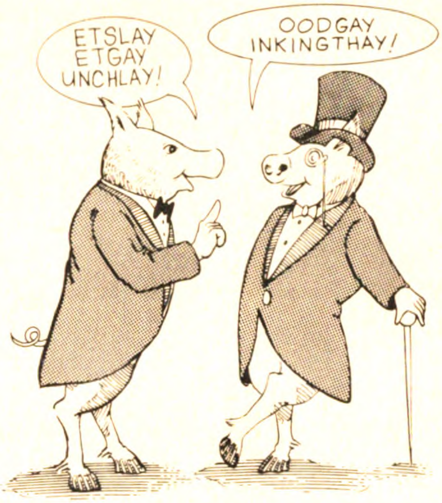

START
OPEN A CARTON OF MILK.
TILT THE CARTON SO THAT MILK POURS INTO A GLASS.
IS THE GLASS FULL?
NO
YES
STOP POURING.
END

Programs and Data

Input, the information you give the computer, is called **data**. (The word *data* means more than one piece of information. Occasionally, you may see the word *datum*, which means one piece of information.)

A **program** tells the computer exactly what to do with the data you give it. The program below tells you how to translate English words into a silly language called Pig Latin.

1 For this program, A, E, I, O, U, and Y are considered vowels.
2 Look at the first word you want to translate.
3 Find the first vowel in the word.
4 If there are any letters that come before the first vowel, take them from the beginning of the word and put them at the end.
5 Add AY to the end of the word.
6 Is there another word to be translated?
  If so, start again at line 3.
  If not, your translation is finished.


ETSLAY ETGAY UNCHLAY!
OODGAY INKINGTHAY!

RUN the program above to find out what your name would be in Pig Latin. Then answer the questions below.

1. When you ran this program, what was your input?

2. What kind of process took place in this program?

3. What was the program's output?

4. What data did the program work on?

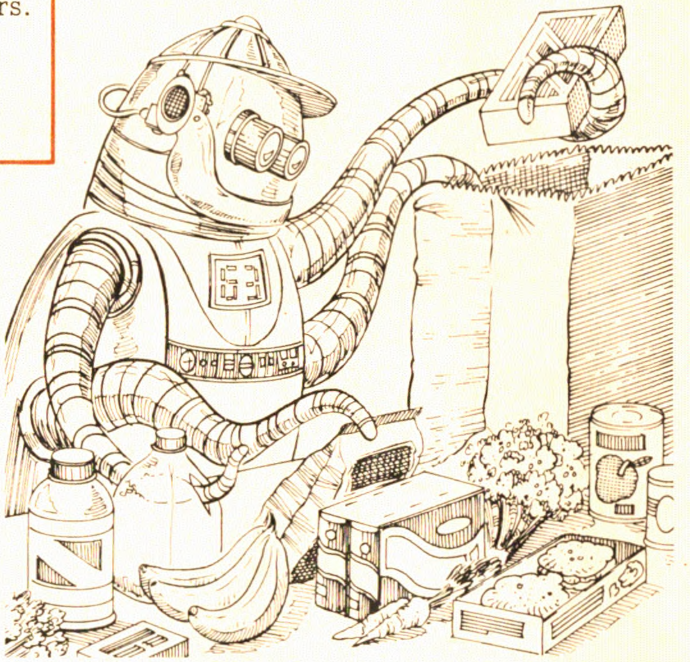
5. RUN the program again. This time, use the name of your favorite movie or television show as data. What is the program's output this time?

Sequence

A computer follows directions step by step, exactly as they appear in a program. When you write a program, you must put the instructions in the right order, or the result won't be what you intended!

The program below tells a robot how to put groceries into a shopping bag.

```
1 Open an empty bag.
2 Put in bread, cookies, and crackers.
3 Put in eggs.
4 Put in fruits and vegetables.
5 Put in cans and bottles.
```



1. What would happen if the robot followed this program?

2. Rewrite the program, putting the steps in a better order.

Below is another mixed-up program for a robot. Put the instructions in the best order by numbering the steps 1 through 6.

_____ Pour the mixture into the pan.

_____ Mix with an eggbeater.

_____ Break two eggs into a bowl.

_____ Cook and stir for a few minutes, until done.

_____ Add a little milk and some salt and pepper.

_____ Melt a little butter in a pan.

1. If the robot followed the directions in the correct order, what would it make?

____

2. What would happen if the robot did step 1 last?

____

3. Choose another step and tell what would happen if the robot left it out completely. If the robot left out step number ______, ______________________

____

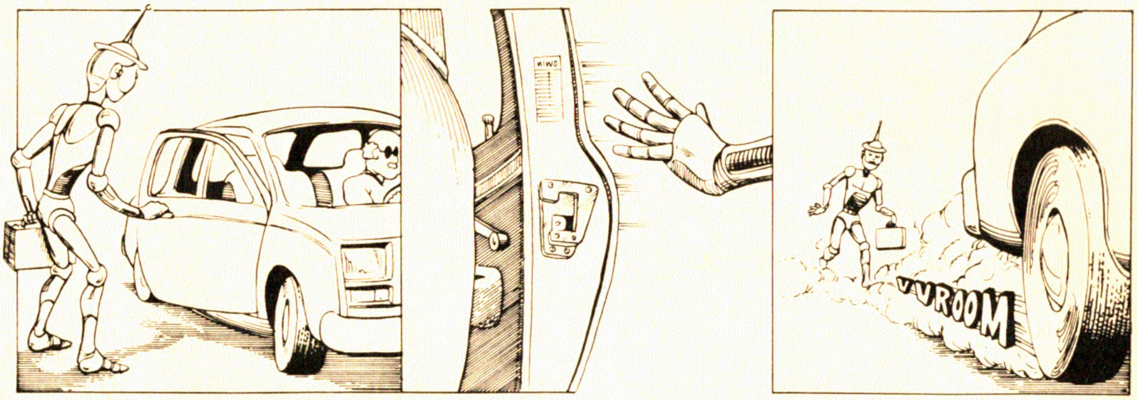
____.

Numbering Lines in a Program

Sometimes when you're writing a program, you accidentally leave out a step. Here, for example, is a program that tells a robot how to get into a car:

```
1 Open the car door.
2 Close the car door.
```

And here is what would happen if the robot faithfully followed the directions:



1. What important statement is missing from the program above?

2. Where should that missing statement appear?

Unfortunately, there's no room to insert a new line without using a strange number such as $1\frac{1}{2}$ or 1.73.

You'll often want to add new lines to a program. An easy way to leave room for new lines is to number the lines in a program by 10s. So the program above should be:

```
10 Open the car door.
20 Close the car door.
```

Then you could number the new line 15, which still leaves room between 10 and 15 and between 15 and 20 in case there are any *more* lines you might want to add.

Imagine that you want to add the following lines to the program. What number would you give each line?

_____ Fasten your seat belt.

_____ If the car door is locked, unlock it.

_____ Make sure your hands and feet are clear of the door.

Now number the lines in the program below, leaving room to add new lines later. (The steps are already in the right order.)

HOW TO CLEAN YOUR ROOM

_____ Make your bed.

_____ Put away your clothes.

_____ Dust your furniture.

_____ Vacuum the floor.

_____ Throw away old papers.

Now number the additional lines below so they'll appear in the right place in the program above. (Remember to leave some room between the new and old lines in case you want to add more lines.)

_____ Pick things up off the floor.
_____ Clear the junk off your bed.
_____ Empty the wastebasket.

Now write your own new line for the program, numbering it so that it will appear in the right place.

____

Running a Program in Order

When you run a computer program, the computer starts with the lowest line number, whether it's 1 or 25 or 200. Then it looks for the next higher line number—even if the lines are mixed up.

Here's a mixed-up program telling how to take a bath.

```
120 Get out of the tub.
 20 When the tub is half full, turn off the water.
 10 Turn on the water.
200 Open the drain to let the water out.
  5 Close the drain.
100 Scrub yourself.
 50 Get into the tub.
```

Use the program above to do the following items.

1. Write the line numbers in the order in which you would do the statements.

2. What would be the best number for each of the following lines?

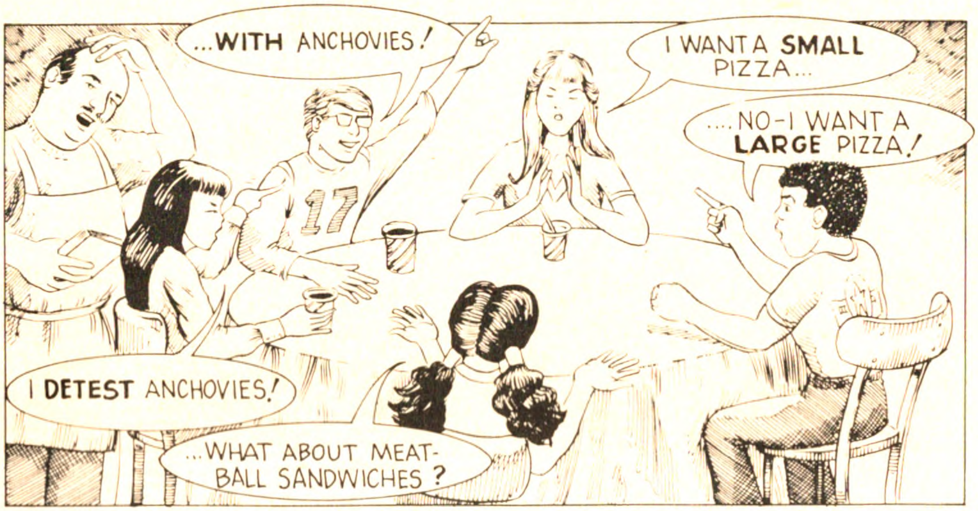
_____ Check the water temperature.

_____ Dry yourself off.

_____ Rinse yourself off.

_____ Take off your clothes.

3. Now write the line numbers in order for the whole program. Include the four new lines you numbered.


...WITH ANCHOVIES!
I WANT A SMALL PIZZA...
....NO-I WANT A LARGE PIZZA!
I DETEST ANCHOVIES!
...WHAT ABOUT MEAT-BALL SANDWICHES?

Here's another mixed-up program. This one tells how to go out and have pizza.

Decide whether you want a small or large pizza.
Eat the pizza.
Wait 10 minutes longer than you were told.
Go to the pizza parlor.
Take the pizza to your table.
Order your pizza.
When the pizza is ready, pay for it.
Argue about whether to have anchovies on your pizza.

Rewrite the statements above in the space below. Put them into the correct order. Then give each statement a line number, starting with 10 for the first line.

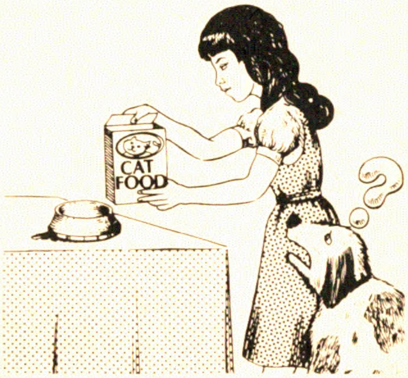
Making Changes

Sometimes you want to make changes in a program. Do you have to write the whole program over just to make a small change? No, there is a much easier way. If you write a new line with the old line number, the new line automatically replaces the old one.

Here's a program that needs some changes.

HOW TO FEED A DOG

10 Get the dog's dish.
20 Get some cat food.
30 Put some food in the dish.
40 Put the dish on the ceiling.
50 Make sure the dog has a bowl of soda pop to drink.
60 Call the dog.



Use the program above to do the following items.

1. How many lines does this program have? ____________________

2. Which three lines should be changed? ____________________

3. Write the three new lines, giving each the number of the line you want it to replace.

____

____

____

4. How many lines will the corrected program have? ____________________

Whenever you write a new line using an old line number, the computer forgets the old line and only remembers the *last* time you use that particular line number.

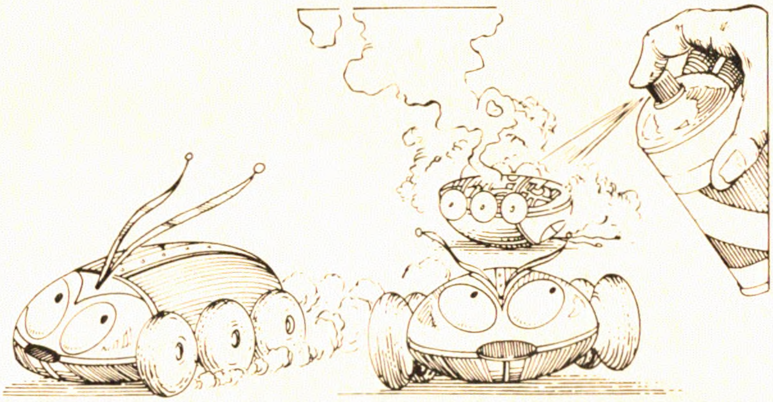
Here is a program that contains a lot of changes:

HOW TO PACK A SUITCASE

```
10 Open a suitcase full of moths.
10 Open an empty suitcase.
20 Decide what you need for your trip.
30 Put your shoes in the suitcase first.
40 Wad up your clothes into little balls.
40 Pour water all over your clothes.
40 Carefully fold your clothes.
50 Put them into your suitcase.
50 Lay them neatly in your suitcase.
60 Close the suitcase.
```

Write the program in the correct order, leaving out any lines that were replaced by new lines.

Debugging



No matter how careful a programmer is, errors somehow always manage to creep into a program. Programmers often call these errors **bugs**, so **debugging** a program means finding and getting rid of the errors in it.

Here are four common kinds of errors:

- *Omission*: An instruction left out of the program. For example, in the instructions, "Plug television in. Turn to channel 2," the instruction "Turn the television on" has been omitted.
- *Unclear instruction*: A direction that isn't specific enough; for instance, "Buy groceries" instead of "Buy one quart of milk and a small jar of peanut butter."
- *Wrong instruction*: A direction that tells exactly what to do but simply isn't correct; for example, "Add ice cream, chocolate syrup, and mustard."
- *Wrong order*: Instructions that are out of sequence; for example, "Put the ice tray into the freezer. Fill the ice tray with water."

Identify the error in each of these programs.

HOW TO WASH YOUR HAIR

```
10  Get your hair wet.
20  Rub your hair with your fingers.
30  Rinse your hair.
40  Put shampoo on your hair.
50  Dry your hair.
```

1. Which type of error does this program contain?

____

2. How would you correct the error?

____

HOW TO DO YOUR HOMEWORK

10 Get your books, notebooks, papers, pens, and pencils together.
20 Turn on the television and radio.
30 Sit at a table or desk.
40 First do the assignments that are due tomorrow.
50 Then work on assignments that are due later.

3. Which type of error does this program contain?

____

4. Write a new line to correct the error.

____

HOW TO BUY CANDY FROM A VENDING MACHINE

10 Decide what kind of candy you want.
20 Put the correct change into the money slot.
30 Press a button.
40 Take your candy from the opening at the bottom of the machine

5. Which type of error does this program contain?

____

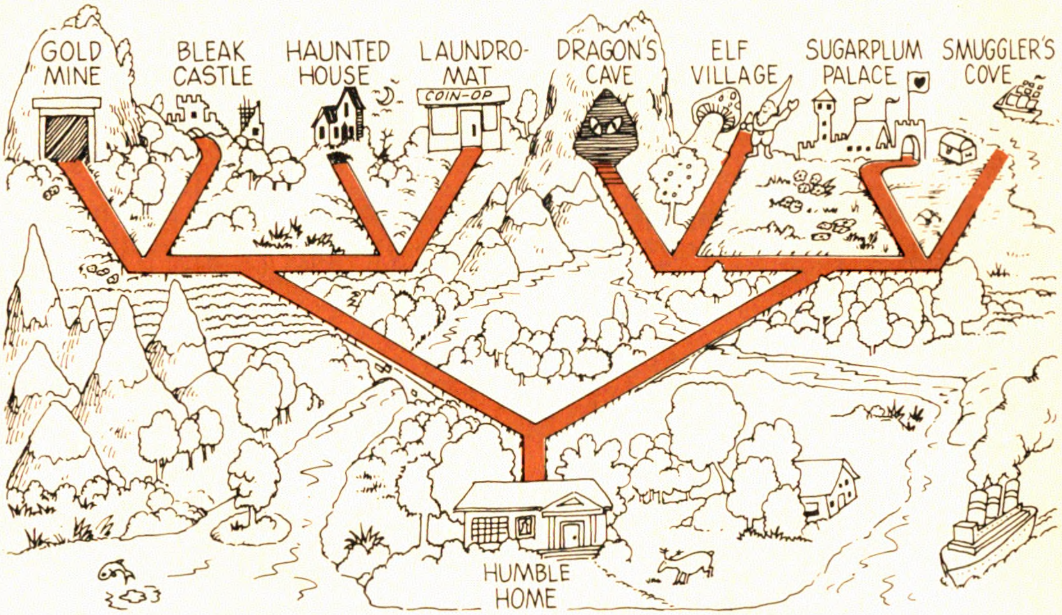
6. Write a new line to correct the error.

____

Branching

When a road gives a choice of different ways to go, it is said to **branch** in different directions.

Here's a map of a road with many branches. If you were walking in this strange countryside, you would have two choices every time the road branched—to turn right or to turn left. To get from *Humble Home* to *Bleak Castle*, for example, you would turn left, then left again, and then right.


GOLD MINE
BLEAK CASTLE
HAUNTED HOUSE
LAUNDRO-MAT
COIN-OP
DRAGON'S CAVE
ELF VILLAGE
SUGARPLUM PALACE
SMUGGLER'S COVE
HUMBLE HOME

If you left Humble Home *and made these choices when the path branched, where would you wind up?*

1. right, left, left ____________________

2. left, left, left ____________________

3. left, right, left ____________________

If you left Humble Home *and wanted to reach the destinations below, what choices would you make?*

4. Elf Village ____________________

5. The Laundromat ____________________

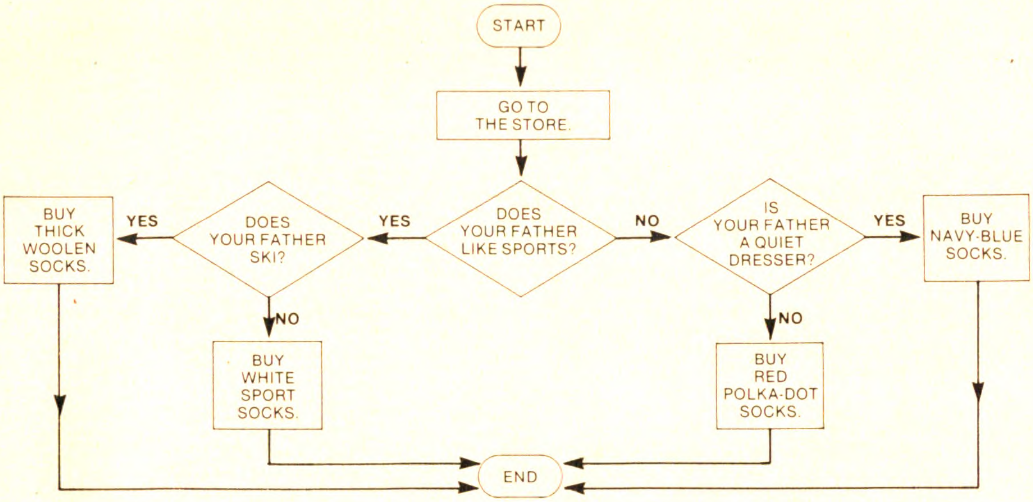
6. Smuggler's Cove ____________________

Branching Programs

When a program gives a choice of different things to do, it, like a road, is said to **branch**.

Here is a flowchart for a program with several branches:

How to Buy Socks for Your Father


START
GO TO THE STORE.
DOES YOUR FATHER LIKE SPORTS?
YES
NO
DOES YOUR FATHER SKI?
YES
NO
BUY THICK WOOLEN SOCKS.
BUY WHITE SPORT SOCKS.
IS YOUR FATHER A QUIET DRESSER?
YES
NO
BUY NAVY-BLUE SOCKS.
BUY RED POLKA-DOT SOCKS.
END

Look at the flowchart above carefully. Then use it to answer the questions below.

1. How many choices do you have each time the program branches?

____

2. How many different kinds of socks does this program help you choose from?

____

3. If you answer the questions in "Yes, No" order, which kind of socks would you buy?

____

4. If you answer the questions in "No, Yes" order, which kind of socks would you buy?

____

5. Which steps must you do no matter which path you follow?

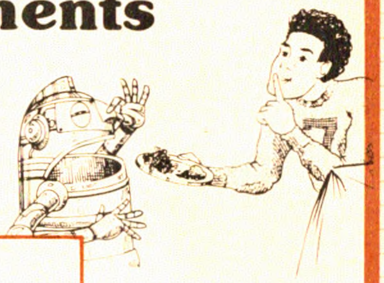
____

____

____

____

IF . . . THEN Statements



A **branching program** gives a choice of things to do. Here is a very simple branching program:

```
10 Sit down to a dinner of liver and spinach.
20 IF you like liver and spinach, THEN eat your dinner.
30 IF you don't like liver and spinach, THEN
   secretly slip your dinner to the robot.
```

Use the program above to answer the following questions.

1. What are the possible results of this program?

____

2. Which statements ask you to make decisions?

____

3. Suppose that you like liver and spinach. What do you do immediately after reading line 20?

____

4. Suppose that you don't like liver and spinach. What do you do immediately after reading line 20?

____

A **decision statement** makes a choice. It has two parts. The part following IF is the **condition** (for example, "you like liver and spinach"). The part following THEN is the **result** (for example, "eat your dinner"). Here's how a decision statement makes a choice: If the condition is *true*, then the result happens. If the condition is *not true*, the result is ignored. For example, IF you do not like liver and spinach, you ignore the result part of line 20 above.

Answer the following questions about "IF . . . THEN" statements.

5. What is the condition in line 30 above? Circle it.
6. What is the result in line 30 above? Underline it.
7. What is the condition in the statement below? Circle the condition.

IF it is raining, THEN put up your umbrella.

8. What is the result in the statement above? Underline the result.

Logic: *And* and *Or*

All **conditions** start with the word IF. But after that, they can take different forms. A simple one is, "IF *it's Monday*, THEN we have leftovers for dinner." Other conditions have two or more parts; for example, "IF *it's Friday*, *and we get home late, and we don't want to cook*, THEN we eat out."

How can you tell whether the whole condition is true?

When the parts are connected by *and*, *all* the parts must be true for the condition to be true. For example:

> IF it is dark and it is clear, THEN we look at the stars.

Use the boxed sentence above to do each item below.

1. Circle the condition. Underline the result.

2. What are the two parts of the condition?

____

3. Suppose it is dark and it is clear. Do we look at the stars? ______________

4. Suppose it is dark, but it is not clear. Do we look at the stars? ______________

5. Suppose it is clear, but it is not dark. Do we look at the stars? ______________

When the parts of a condition are connected by *or*, the condition is true if *any* part is true. For example:

> IF it is cold or it is rainy, THEN we stay home.

Use the boxed sentence above to do each item below.

6. Circle the condition. Underline the result. ______________

7. What are the two parts of the condition? ______________

8. Suppose it is rainy and cold. Do we stay home? ______________

9. Suppose it is cold but not rainy. Do we stay home? ______________

10. Suppose it is not cold and not rainy. Do we stay home? ______________

Logic: A Quick Quiz

Remember these rules about decision statements:

- If the condition is true, the result is true.
- When parts of a condition are connected by *and*, the condition is true only if *all* the parts are true.
- When parts of a condition are connected by *or*, the condition is true if any part of the condition is true.

Read each statement below. Based on the information given, decide if the result is true. Write yes in the blank if it is true; no, *if it is not true.*

1. IF the pond is frozen, THEN you go ice skating.

The pond is frozen. Do you go skating? ______

2. IF you get more than 80 on the test, and you do all your homework, THEN you get an A.

You get a 90 on the test, and you do all your homework. Do you get an A? ______

3. IF the oven is too hot or the cookies bake too long, THEN the cookies burn.

The oven is just right, but you leave the cookies in too long. Do the cookies burn? ______

4. IF your homework is done, and your room is clean, and you have enough money, THEN you can go to a movie.

Your homework is done, your room is clean, and you are broke. Can you go to a movie? ______

5. IF you are ten years old or older, THEN you have celebrated your decennial.

Have you celebrated your decennial? ______

6. IF you like chocolate, and you like hamburgers, THEN you like chocolate-covered hamburgers.

You like chocolate, and you like hamburgers. Do you like chocolate-covered hamburgers? ______

7. IF $3 \times 4 = 12$ and $4 + 3 = 8$, THEN you win a prize.

Do you win a prize? ______

8. IF you smell smoke, or you see flames, or you hear a fire alarm, THEN leave the building.

You smell smoke and hear a fire alarm, but you don't see flames. Do you leave the building? ______

GO TO . . .

You already know how to show **branches** in a flowchart.

But how do you make a **branch** in a program? You can use a **decision statement** to send the computer to the right set of instructions, like this:

```
10  Go to the kitchen.
20  You can choose to drink water or cola.
30  If you want cola, then GO TO line 70.
40  Get a glass.
50  Fill it with water.
60  GO TO line 90.
70  Open a bottle of cola.
80  Stick a straw into the bottle.
90  Drink.
```

Line 30 is a **decision statement**. The words *GO TO* tell you to skip to a different place in the program where you'll find further instructions. That's how the program **branches**.

Answer the following questions about the program above.

1. What is the condition in line 30?

____

2. Suppose that you want to drink cola. Which line do you read after you read line 30?

____

Which lines do you skip?

____

3. In order, write the line numbers of all the instructions you follow if you want cola. (Leave out the lines you skip.)

____

4. Suppose you want water. Which line do you read after you read line 30?

____

Which line do you read after you read line 60?

____

Which lines do you skip?

____

5. In order, write the line numbers of all the instructions you follow if you want to drink water.

____

GOTO

In real computer programs, you may see the words *GO TO* made into one word, like this: GOTO. For example:

```
10 GOTO 80
```

Line 10 tells the computer to skip to line 80, where it will find further instructions.

Here is a branching program with some incomplete GOTO statements. Read the whole program, then fill in the missing line numbers.

HOW TO MAKE A SUNDAE

```
 10 IF you like chocolate THEN GOTO ____.
 20 IF you like strawberries THEN GOTO ____.
 30 For a pineapple sundae, put vanilla ice
    cream in a dish.
 40 Pour pineapple sauce over the ice cream.
 50 GOTO 110.
 60 For a special chocolate sundae, put
    chocolate ice cream in a dish.
 70 Pour chocolate sauce over the ice cream.
 80 GOTO ____.
 90 For a strawberry sundae, put strawberry
    ice cream in a dish.
100 Pour strawberry sauce over the ice cream.
110 Put whipped cream on top.
120 Serve your sundae.
```

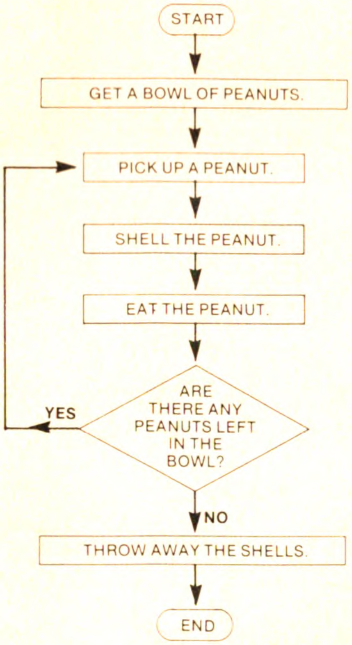
1. Suppose you like strawberries. Which lines would you skip as you go through the program? ____________________

2. Which lines would you read no matter which flavor you prefer? That is, which lines do *all* the branches have in common? ____________________

Loops

A **loop** in a flowchart is easy to see.

Draw a circle around the loop in the flowchart.


START
GET A BOWL OF PEANUTS.
PICK UP A PEANUT.
SHELL THE PEANUT.
EAT THE PEANUT.
ARE THERE ANY PEANUTS LEFT IN THE BOWL?
YES
NO
THROW AWAY THE SHELLS.
END

Programs use the instruction GOTO to make loops. Here is the program for the flowchart.

Draw a circle around the loop in the program.

```
10 Get a bowl of peanuts.
20 Pick up a peanut.
30 Shell the peanut.
40 Eat the peanut.
50 IF there are any peanuts left
   in the bowl, THEN GOTO 20.
60 Throw away the shells.
70 END.
```

Use the flowchart and the program to answer the following questions.

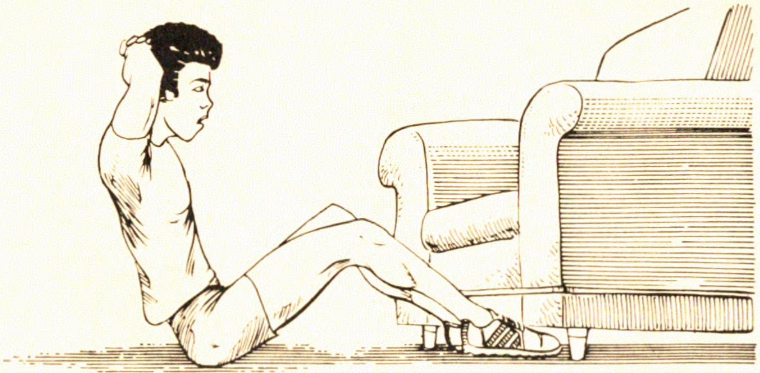
1. Which line in the program does the same job as the diamond-shaped decision step in the flowchart? ______
2. If you reach line 50, and there *are* peanuts left, what do you do? ______
3. If you reach line 50, and there *aren't* any peanuts left, what do you do? ______
4. How many times do you perform the loop? ______
5. Which lines are performed only once? ______

Endless Loops

Loops are useful whenever you write a program with a task that is repeated over and over. For example, imagine a person following this program:

HOW TO DO SIT-UPS

```
10 Lie flat on your back with your
   knees slightly bent.
20 Keeping your feet on the floor,
   sit up.
30 Lie down again.
40 GOTO 20.
50 END.
```



1. Which lines are included in the loop? ______________________________

2. How many times would the person sit up if he or she followed this program?

   ____

3. The loop in this program is called an **endless** or **infinite loop**. Before long, the person following the program would get very tired! Here is a statement that lets the person escape from the loop:

   IF you are tired, THEN GOTO 50.

   What should this statement's line number be? ______________________

4. There is a second way to let someone escape from the loop. The result is the same, but instead of adding a line, you change line 40 so that only a person who is *not* tired will continue to do sit-ups. Complete this new line 40:

   40 IF ______________________________ THEN GOTO 20.

Here is another program that contains an endless loop.

HOW TO READ A BOOK

```
10 Open the front cover.
20 Turn to page one.
30 Read page one.
40 Go on to the next page.
50 Read that page.
60 GOTO 40.
70 END
```

5. Which lines are included in the loop?

6. Anyone who follows this program will eventually reach the end of the book. Write a decision statement that tells the person to skip to the end of the program if he or she has read the last page of the book. (HINT: The statement should be line 55.)

7. There is a second way to let someone escape from the loop. Write a new line 60 that tells the person to continue the loop only if he or she has not read the last page of the book.

8. A computer has limited memory space, so the shorter the program, the better. Which way of escaping from the loop takes the fewest lines?

Debugging: Branches and Loops

Here are three kinds of errors that often show up in branches and loops. Can you correct them?

- *IF without THEN*: A decision statement starts with IF to introduce the *condition*, and it must have the word THEN to introduce the *result* part of the statement. THEN is missing from this line:

```
10 IF your name is Lily, GOTO 30.
```

1. What is the condition in this statement?

___

2. What is the result?

___

3. Rewrite the statement with THEN in the correct place.

___

- *GOTO the wrong line*: A simple mistake like a wrong line number can cause big problems. For example, line 10 of this program tells you to GOTO the wrong line:

```
10 IF you feel tired, THEN GOTO 20.
20 Do some exercises.
30 GOTO 50.
40 Take a nap.
50 END
```

4. Does this program contain a loop or a branch? ___

5. Suppose you feel tired. What do you do after you read line 10 ?

6. What *should* line 10 say?

■ *Endless loop*: Loops are useful for jobs that need to be repeated many times. But a program shouldn't repeat a task forever! It's important to tell when to *stop* repeating a loop. Like a flowchart, a program must lead from the first line to the line that says END. For example, this program has an endless loop:

```
10  Stand up.
20  Turn to your right.
30  GOTO 20.
40  END
```

7. How many times would you repeat this loop?

8. Is it possible to reach line 40?

9. One way to provide an escape from the loop is to change line 30. Write a new line 30 that tells you to keep turning only if you aren't dizzy.

10. A second way to provide an escape from the loop is to add a line saying: IF you are dizzy, THEN GOTO 40. What *should* this statement's line number be?

Something to think about: Which solution makes the shortest program?

Review: PROGRAMS

If you look carefully, you can find all of the things in the list below in the program that follows the list.

branch
loop
decision statement
result
condition

HOW TO CHOOSE A PIECE OF CHOCOLATE CANDY

```
10  Take a piece of candy from the box.
20  Use your thumb to poke a little hole
    in the bottom of the candy.
30  Look carefully at the filling.
40  IF you like the filling, THEN GOTO 70.
50  Put the candy back into the box so
    the hole doesn't show.
60  GOTO 10.
70  Eat the candy.
80  END
```

Use the words in the list at the top of the page to fill in the blanks in each of the following sentences:

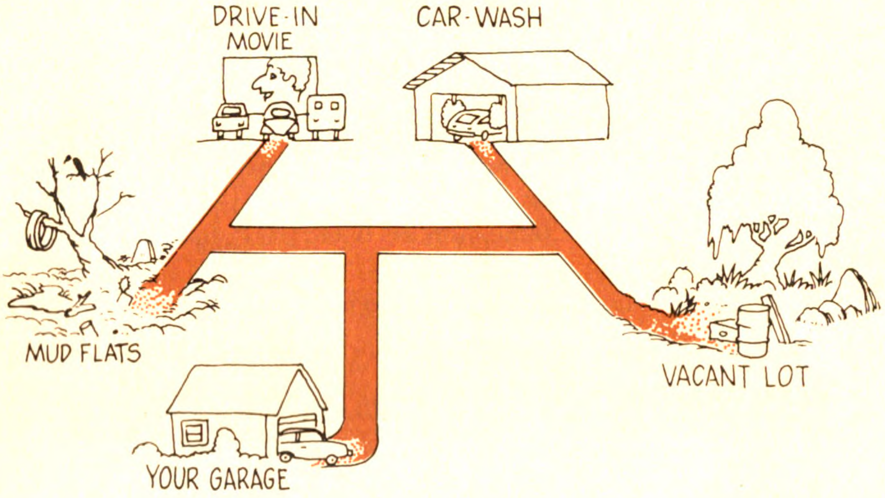
1. Line 40 is a ______________________________.

2. Line 40 gives a choice. It makes a ____________________ in the program.

3. In line 40, "you like the filling" is the ______________________________.

4. In line 40, "GOTO 70" is the ______________________________.

5. Lines 10 through 60 make a ______________________________.


DRIVE-IN MOVIE
CAR-WASH
MUD FLATS
VACANT LOT
YOUR GARAGE

The branching program above tells how to get from your garage to either the drive-in movie or to the car wash.

Fill in each blank below to complete the program.

```
 10  Drive out of your garage.

____ IF you want to go to the drive-in movie,

     THEN GOTO _______________.

____ Turn right.

____ Turn left.

____ GOTO 80.

 60  Turn left.

____ Turn _______________.

 80  Pay the attendant.

____ Drive in.

100  _______________
```

4 CODES

In Other Words . . .

One idea or number can be expressed in many different ways. Two different words for the same idea—for instance, *lucky* and *fortunate*—are called *synonyms*. Two different ways of saying the same number: 5 and 3 + 2, for example, are called **equivalent expressions.**

Computer programmers use the = sign to mean "equal, the same." The sign <> means "less or greater than."

Think about each pair of items below. If both expressions mean exactly the same thing, write = in the box. If they mean different things, write <>.

1. 4 + 6 + 13	☐	24
2. October	☐	the ninth month of the year
3. Carla's mother's brother's daughter	☐	Carla's cousin
4. never	☐	not always
5. input	☐	opposite of output

Now write two different expressions that mean the same thing as each expression below.

6. 10 = ____________________ = ____________________

7. one year = ____________________ = ____________________

8. seldom = ____________________ = ____________________

9. zero = ____________________ = ____________________

Something to think about: How many different ways are there to express a number such as 3? ____

A computer can compare two numbers and tell which is bigger.

Compare the numbers below and write > , < , *or* = in the boxes.

Remember:
> means "larger than"
< means "smaller than"
= means "equal to"

1001 ☐ 956

1,000,000 ☐ 100,000

$\frac{1}{2}$ ☐ $\frac{2}{4}$

But often the computer must do some work to the two numbers before it can compare them.

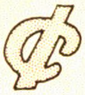
For each pair below, first figure out the two numbers. Then write them in the blanks. Next, compare the two numbers and write > , < , *or* = *in the box. The first one has been done.*

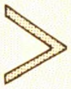
10.	27	3 × 9	<	4 × 7	28
11.	______	Number of days in February in a leap year	☐	Two dozen	______
12.	______	Number of dwarfs met by Snow White	☐	Number of hours from 10 a.m. to 3 p.m.	______
13.	______	Number of states in the U.S.A.	☐	1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10	______
14.	______	Number of hours in three days	☐	Number of pages in this book	______
15.	______	Number of letters in the word *antidisestablishmentarianism*	☐	Number of nickels in a dollar plus number of wheels on four bicycles	______
16.	______	Number of legs a spider has times number of sides a cube has	☐	Number of weeks in a year minus number of eyebrows on the President of the United States	______

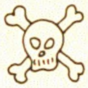
Symbols and Codes

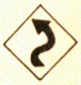
A **symbol** is something that stands for another thing or idea. For example, the symbol + means "plus."

What does each of the following symbols mean? Match each symbol with its meaning by writing the letter of the meaning in the blank next to its symbol.

____ **1.**


____ **2.**


____ **3.**


____ **4.**


a. curve in the road
b. poison
c. cent
d. greater than

A **code** uses symbols for the letters or words in a message. Here is a code that uses numerals as symbols for letters:

A = 1
B = 2
C = 3
D = 4
E = 5
F = 6
G = 7
H = 8
I = 9
J = 10
K = 11
L = 12
M = 13
N = 14
O = 15
P = 16
Q = 17
R = 18
S = 19
T = 20
U = 21
V = 22
W = 23
X = 24
Y = 25
Z = 26

In the code above, 19 stands for *S*, and 19-25-13-2-15-12 stands for the word *symbol*.

Decode the following messages.

5. 13-5-5-20 13-5 1-20 20-5-14.

____

6. 4-5-19-20-18-15-25 20-8-9-19 14-15-20-5.

____

Write the messages below in code.

7. Hi. ____________________

8. Come to a party. ____________________

9. (Your name.) ____________________

Something to think about: How could you use this code to write numbers instead of words? Write your telephone number in code.

____

Variables

A **variable** is a symbol—usually a letter—that stands for a number or *value*. For example, if we say that D = 12, then the value of the variable D is 12. You can assign a value to a variable by saying, *"LET D = 12."*

What is the value of each of the following variables?

1. LET M = number of blind mice M = ____________

2. LET H = number of halves in a whole H = ____________

3. LET W = number of days in a week W = ____________

4. LET A = day on which April Fool's Day falls A = ____________

5. LET S = 3 + 2 S = ____________

Use the variables above to calculate the values of these variables:

6. LET X = M + W X = ____________

7. LET B = H − A B = ____________

8. LET L = S × 4 L = ____________

9. LET J = W + H J = ____________

10. LET T = A + M + W + 2 T = ____________

Variables are a way to store information in a computer. Once you assign a value to a particular letter, the computer will remember that value until you assign a *new* value to the letter. You can change the value as often as you like, which is probably why these letters are called variables—you can *vary* their value.

What are the different values of Y below? LET Y = year

11. What is the value of Y this year?

____

12. What is the value of Y for when you were born?

____

13. What is the value of Y for your tenth birthday?

____

14. What is the value of Y for the last day of this century?

____

Strings

Sometimes the information you want a computer to remember is a word instead of a number. A word or any series of letters is called a **string**. For example, "Josephine" is a string that is nine letters long.

You can store strings in a computer's memory in much the same way as you store numbers. The symbol you use is a letter—just as you would for a number—but you also add a dollar sign, which is the symbol for "string." So if you tell the computer: LET Z$ = "JOSEPHINE"

the computer will remember that the symbol Z$ stands for the word between quotation marks – JOSEPHINE – until you tell it otherwise.

Here are some examples of strings:

A$ = "A"
B$ = "ACT"
C$ = "AND"
D$ = "ANYWHERE"

E$ = "CLIMB"
F$ = "GARBAGE"
G$ = "HE"
H$ = "HIS"

I$ = "HOLD"
J$ = "LIKE"
K$ = "NUT"
L$ = "PANTS"

M$ = "TO"
N$ = "TREE"
O$ = "TRUCK"
P$ = "UP"
Q$ = "WANTS"

The answers to the riddles below are in a code that uses the strings above. Decode each answer.

1. Where does a grizzly bear sleep?

   D$ G$ Q$ M$ ______________________________

2. What has four wheels and flies?

   A$ F$ O$ ______________________________

3. Why does a fireman wear red suspenders?

   M$ I$ H$ L$ P$ ______________________________

Strings don't have to be whole words. They can include any series of letters and spaces. For example, if A$ = "H" and B$ = "I, JOE!" then A$ plus B$ equals "HI, JOE!"

Here are some strings that, at first, don't seem to make sense:

A$ = "E JE"
B$ = "D HI"

C$ = "EANS"
D$ = "SO H"

E$ = "LLY B"
F$ = "DE I"

G$ = "E COUL"
H$ = "N TH"

Decode the answer to the riddle below by using the strange strings above. (Don't forget to include all the spaces.)

4. Why did the elephant paint his toenails different colors?

   D$ G$ B$ F$ H$ A$ E$ C$ ______________________________

   ____

PRINT

Once you have stored numbers or words (or both) in a computer's memory, it's easy to tell the computer to show them on its monitor or to print them on paper. Suppose you have told the computer: `LET Z$ = "JOSEPHINE"`

Now you have two ways to tell the computer to print the word "Josephine." You can say: `PRINT "JOSEPHINE"` *or:* `PRINT Z$`

The results are exactly the same. Here's what you would see on the video monitor or printed on paper: `JOSEPHINE`

(**Note**: Don't put quotation marks around a variable—like A—or a string variable—like Z$—unless you want the computer to print the symbol itself, like this: Z$)

Here is a program that would run on a real computer:

```
10 LET A$ = " WERE"
20 LET B$ = " GREEN"
30 LET C$ = " WASH"
40 LET D$ = " HER"
50 LET E$ = " IN"
60 PRINT "WHY DID THE ELEPHANT WEAR RED SOCKS?"
70 PRINT "BECAUSE" D$ B$ " ONES" A$ E$ " THE" C$
80 END
```

Use the program above to answer the following questions.

1. What would the computer print after it read line 10?

____

2. What would the computer print after it read line 60?

____

3. What would the computer print after it read line 70?

____

4. Write exactly what the computer would print if line 70 said `PRINT "BECAUSE D$ B$ ONES A$ E$ THE C$."`

____

5. In lines 10 through 50, why is there a space between the quotation marks and the beginning of each word?

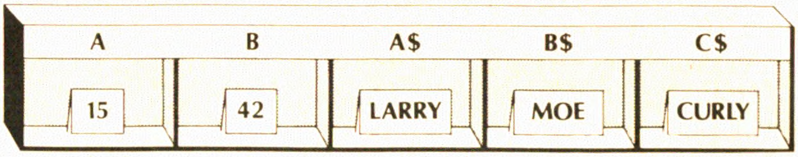
____

Making More Storage Space

You can think of a computer's memory as made up of many small compartments. Each compartment contains one piece of information or data, and each compartment has its own name. For example, if you tell the computer:

```
10  LET A   = 15
20  LET B   = 42
30  LET A$ = "LARRY"
40  LET B$ = "MOE"
50  LET C$ = "CURLY"
```

then you could imagine that the computer's memory looks like this:



1. What is the "name" of the compartment that contains the string LARRY?

2. If you kept filling compartments this way—

(a) how many separate numbers could you store in the computer's memory at A through Z?

(b) how many strings could you store in the computer's memory at A$ through Z$?

Storing numbers and strings this way is useful—if you don't have too many numbers and strings. But what if you want to store more numbers or more strings than there are letters in the alphabet?

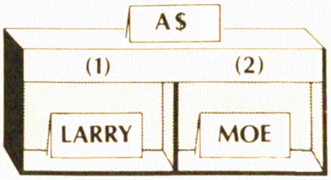
One way to make more storage space is to name compartments using combinations of letters *and* numbers. An **array** is a letter variable which is divided into separate storage spaces. The name of each storage space is the letter variable with a number in parentheses. For example, if you tell the computer

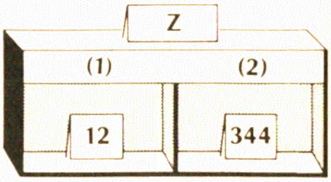
```
10  LET A$(1) = LARRY
20  LET A$(2) = MOE
```

or

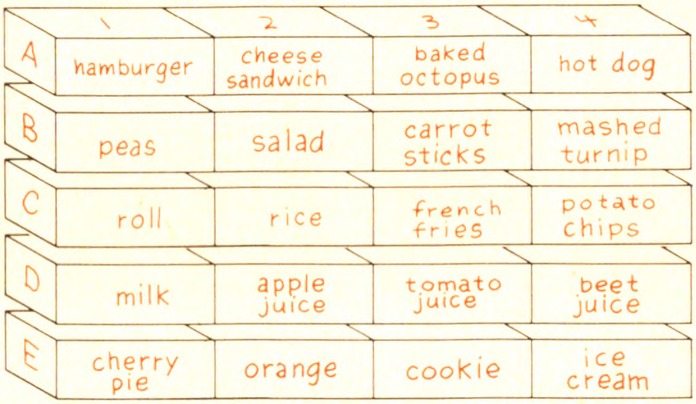
```
10  LET Z(1)   = 12
20  LET Z(2)   = 344
```

you could imagine the computer's memory has two arrays like this:





Here's another way to imagine the possible combinations of letters and numbers. This is the Computerized Lunch-O-Mat. People just type in the letter and number of any food, and it comes sliding out the slot at the bottom.



	1	2	3	4
A	hamburger	cheese sandwich	baked octopus	hot dog
B	peas	salad	carrot sticks	mashed turnip
C	roll	rice	french fries	potato chips
D	milk	apple juice	tomato juice	beet juice
E	cherry pie	orange	cookie	ice cream

3. The foods in each row across have something in common. What are all the foods in row B?

4. What are all the foods in row D?

5. What would you get if you typed the following letters and numbers? A(3), B(4), E(1)

6. What would you type if you wanted a cheese sandwich and a glass of beet juice?

7. How many different foods can you ask for by using the combinations of five letters and four numbers?

8. If a column 5 of five new foods were added to Lunch-O-Mat, how many foods could you ask for all together?

9. If you kept adding lettered rows, how many could you possibly have all together?

10. If the Lunch-O-Mat could be as large as you wanted, how many different combinations of letters and numbers could you possibly have?

Arrays

The Lunch-O-Mat on page 63 had five **arrays** of foods. Here are three *arrays* of data:

	1	2	3	4	5
H	60	62	58	80	
W	112	88	96	200	
A	12	15	13	19	

H stands for height (in inches), W for weight (in pounds), and A for age (in years). You can see all this information about four different people. For example, W(3) = 96 means that the weight of person number 3 is 96 pounds.

1. You are person number 5. Fill in the data for yourself. (If you aren't sure about your height or weight, take a guess.)

2. Write in the values of the following variables:

a. H (3) = ______

b. W(1) = ______

c. A (5) = ______

3. How much would all five people weigh together? Would it be safe for all of you to get on an elevator whose limit was 600 pounds? Here's how to find out:

```
LET X = W(1) + W(2) + W(3) + W(4) + W(5)
```

What is the value of X? X = ______

If X > 600, then stay off that elevator!

Using Arrays

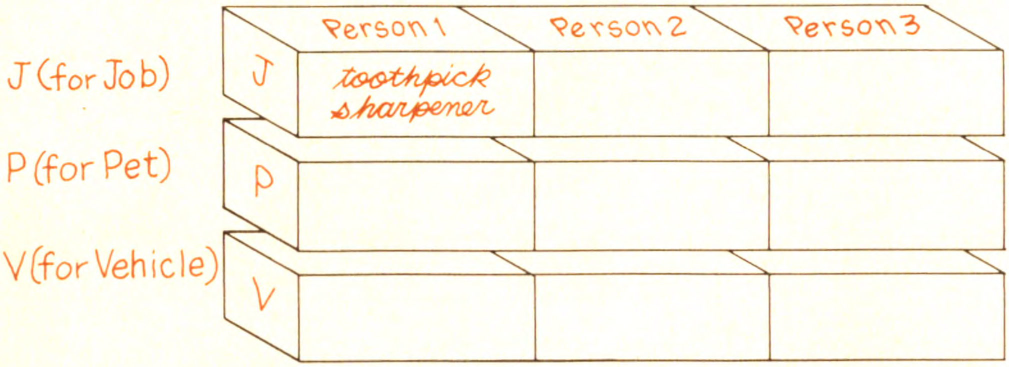
Here is a problem that *looks* difficult. But an array can help you solve it.

You've just met three people. Each one has a different job, a different pet, and a different way of getting around. You know these facts:

1. The toothpick sharpener owns an ostrich.
2. The mattress tester lives next door to the person who rides a skateboard.
3. The owner of the ant always rides a skateboard.
4. The alligator owner hates to fly in his friend's helicopter.

Who is the banana farmer? Who rides a unicycle?

How can you solve this problem? Start by setting up a table with three *arrays* for the data, like this:



		Person 1	Person 2	Person 3
J (for Job)	J	toothpick sharpener		
P (for Pet)	P			
V (for Vehicle)	V			

Read the problem again so you'll know what the three jobs, three pets, and three vehicles are.

Now start with the first clue. Call the toothpick sharpener Person 1. So J(1), Person 1's job, is "toothpick sharpener." J(1) is filled in for you. The clue tells you that Person 1's pet is an ostrich, so write "ostrich" in P(1). Try using the other clues to fill in the array by yourself. If you need more hints, read on.

Go on to the second clue. Call the mattress tester Person 2. Fill in J(2). By the process of elimination, you can fill in J(3). The clue also tells you that the mattress tester does not ride a skateboard, so make a note that V(2) is *not* "skateboard." Read on only if you really need more hints.

The third clue says that the ant owner rides a skateboard. Person 1 owns an ostrich, so V(1) is not "skateboard." We know that V(2) is not "skateboard" either. So Person 3 *must* be both the ant owner and the skateboard rider. Fill in P(3) and V(3). Do you still need hints? Try again before you read on.

By elimination, you can fill in P(2). Now use the fourth clue to decide who rides in a helicopter and who rides a unicycle.

Who is the banana farmer? ______________________________

Who rides a unicycle? ______________________________

Using Variables in Programs

A **variable** can stand for many different pieces of data. When you want a program to use the same *process* on different pieces of data, you don't have to write a different program for every piece of data. Just use variables to stand for the data that will change.

Here is a program that tells you about how many times you breathe in one day. Count your breaths for exactly one minute. (In and out counts as one breath.)

```
10 LET  M = the number of breaths in one minute
20 LET  H = M × 60
30 LET  D = H × 24
40 PRINT "Number of breaths in one day: " D
50 END
```

You be the computer. RUN this program. Fill in the value of each variable below.

1. M = ______________________________

2. H = ______________________________

3. D = ______________________________

4. What gets printed out at the end of the program?

____

5. Which variable stands for the number of breaths in one hour?

____

6. Which variable stands for the number of breaths in one day?

____

7. Now ask three friends how many times they breathe in one minute. RUN the program for each person, and fill in the table or *array* below. (You are Person 1; your friends are 2, 3, and 4.)

	Person 1	Person 2	Person 3	Person 4
M				
H				
D				

8. What is the value of M(3)? M(3) = ____________________

This is the number of times ____________ breathes in one __________.

9. What is the value of H(2)? H(2) = ____________________

This is the number of times ____________ breathes in one __________.

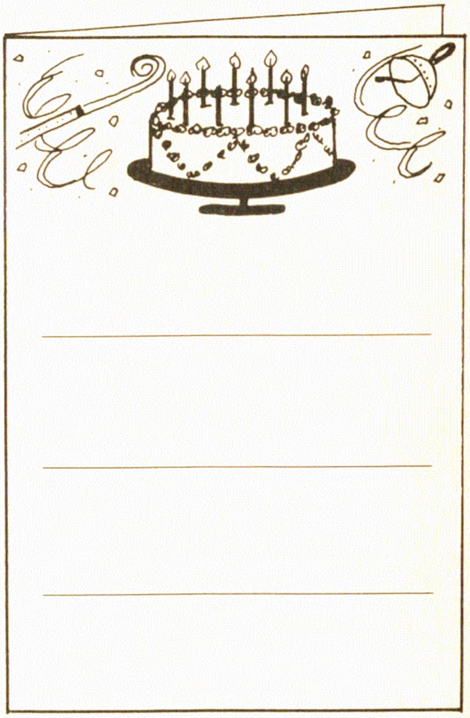
10. What is the value of D(4)? D(4) = ____________________

This is the number of times ____________ breathes in one __________.

Review: CODES

Here is a program that uses strings and variables to help you make a birthday card for a friend.

```
 10  LET A$ = " BIRTHDAY, "
 20  LET B$ = "FROM "
 30  LET C$ = "HAPPY"
 40  LET D$ = "YOU'RE "
 50  LET E$ = your friend's name
 60  LET F$ = your name
 70  LET A  = the year your friend
              was born
 80  LET B  = this year
 90  LET C  = B - A
100  PRINT D$, C
110  PRINT C$, A$, E$
120  PRINT B$, F$
130  END
```



- A **variable** represents a number.
- A **string variable** represents a series of letters.

Use the program above to do the items below.

1. Choose the friend who will receive the card. For your friend, what will E$ represent? E$ = ______________________

2. For your friend, what is the value of A? A = ______________________

3. RUN the program to fill in the birthday card.

5 ACCESS TO COMPUTERS

Commands and Statements

You can give a computer two different kinds of instructions: **commands** and **program statements**. What's the difference?

Commands

- are not part of a program.
- do not have line numbers.
- tell the computer to do something right away.
- include instructions such as RUN and SAVE.
- can also include instructions such as PRINT, if they're not part of a program.

For example: If you type

```
PRINT "JOE"
```

the name *Joe* appears on the computer's monitor right away.

Program statements

- are part of a program.
- have line numbers.
- don't get done until you tell the computer to RUN the whole program.

For example: If you type

```
10 PRINT "JOE"
```

nothing at all will happen until you tell the computer to RUN the program.

Look at each computer instruction below. Decide whether it is a command *or a* program statement. *Write the answer in the blank at the right.*

1. 110 GOTO 340 ________________

2. PRINT "STATEMENT" ________________

3. 5 PRINT "COMMAND" ________________

4. 50 LET Z(1) = 40 ________________

5. RUN ________________

6. 200 END ________________

7. LIST ________________

8. PRINT 4 + 5 ________________

9. 30 PRINT 4 + 5 ________________

10. 90 IF X = 2 THEN GOTO 20 ________________

Writing Programs

A **program** is a set of instructions telling a computer how to do a job. How does a program that you write get into the computer?

- When you write a program, you type it, line by line, on the computer's **keyboard.**
- The first word you type is the command NEW. This isn't part of the program; it just tells the computer that you are about to put in a new program, and it erases any old program inside the computer.
- Then you start typing the **program statements**. Each statement must start with a line number. The computer remembers every statement you type, but it won't actually follow the instructions until later. The last statement in the program says END.
- Now if you type the command RUN, the computer will **execute** your program—that is, it will follow the instructions in the program, line by line.

Here is a program that tells how to write a program for a computer. Fill in the blanks.

10 Sit in front of the computer.

20 Get ready to type all of your instructions on the ________________.

30 To prepare the computer for your program, type the word __________.

40 Type the first line number and ______________________________.

50 Type the next line number and program statement.

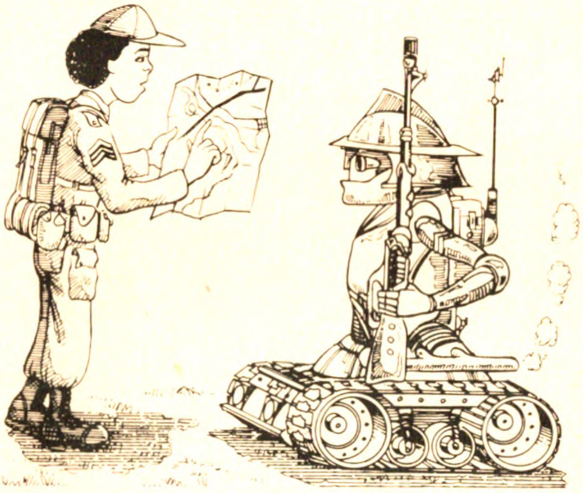
60 If you want to add more instructions then GOTO 50.

70 Type the last line number and the word ______________________.

80 When you're ready for the computer to execute the program, type the command ____.

90 ____

Commands



Once you've typed a program into the computer, **commands** tell the computer what to do with the program. Commands aren't part of a program. They don't have line numbers because they're instructions that the computer follows as soon as you type them.

Match each command in the column on the right with its definition. Write the letter of the correct command in the blank.

_____ **1.** Tells the computer to record the program you've written onto a disk or cassette. This saves the program so you can use it again any time. When you use this command, you also give your program a name so that the computer can file it and find it later.

_____ **2.** Tells the computer to show you a list of all the program statements in your program. The list is in numerical order. Use this command when you want to look back at your program to make changes.

_____ **3.** Tells the computer to follow all the instructions in your program, line by line. That is, it tells the computer to execute your program.

_____ **4.** Tells the computer to find a particular program that has been recorded on a disk or cassette and loads the program into the computer's own memory, ready for use.

_____ **5.** Be careful! This command tells the computer to get ready for a new program—and it erases the old program unless you've saved it on a disk or cassette.

a. LIST

b. SAVE

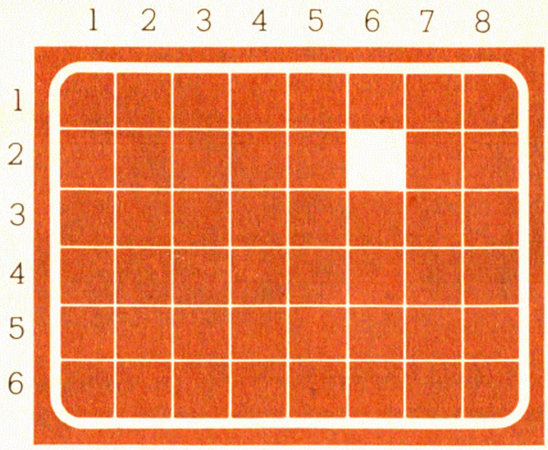
c. LOAD

d. RUN

e. NEW

Graphics

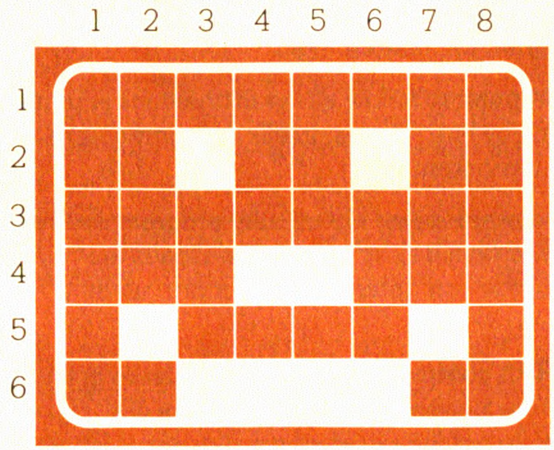
Pictures on a video monitor—or printed on paper by a computer—are called **graphics**. A computer screen is divided into tiny squares. You might imagine the rows and columns on the screen numbered like this:



(A real screen has many more squares, of course.)

When you want the computer to make a particular square white, you tell the computer the square's number *across* (the column number) and then the number *down* (the row number). For example, the white square above is in the sixth column across and in the second row down, so it is square 6,2.

Tell the computer which squares to make white to create the design below.



_____ _____

_____ _____

_____ _____

_____ _____ _____ _____

Fill in each of the following squares to make a picture on the grid below.

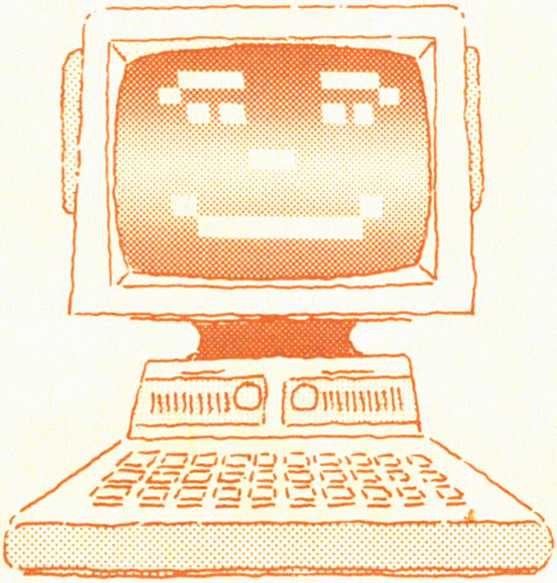
4,8 6,5 7,1 11,3 8,7 9,5 5,7 1,5 3,6 7,2 13,5
5,8 7,3 8,3 10,6 9,6 9,1 7,7 7,5 11,8 6,4 6,3
5,6 6,6 8,8 5,5 12,5 12,6 12,7 11,7 11,6 7,4 2,5
3,5 3,7 4,7 6,8 7,8 5,3 5,4 4,5 2,6 4,6 7,6 6,7
9,8 10,8 9,7 9,4 10,3 13,6 14,5 11,4 10,4 9,2 8,4
8,5 9,3 11,5 10,5 8,6 10,7

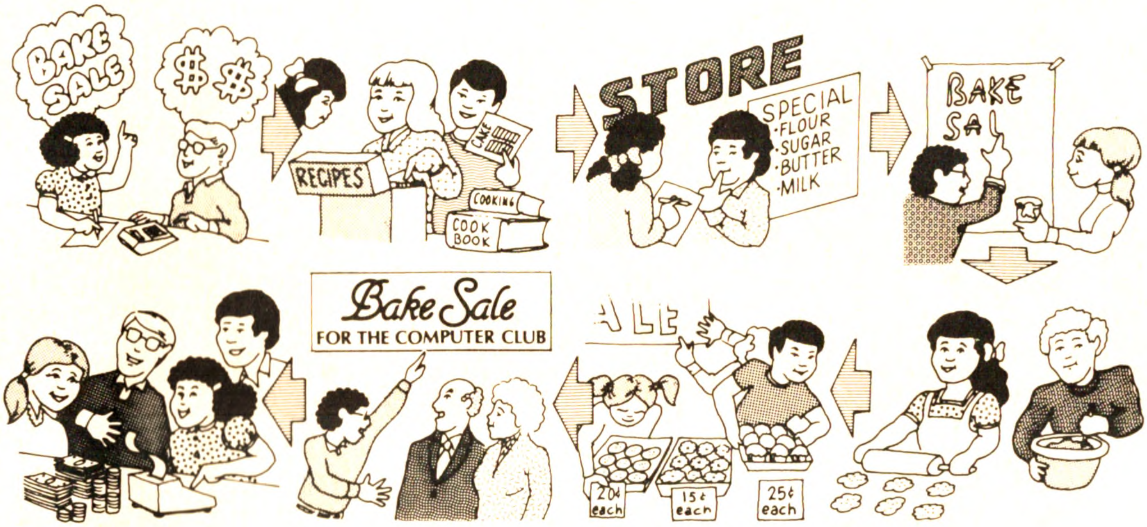
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1														
2														
3														
4														
5														
6														
7														
8														

Uses of Computers

Computers can do dozens of different kinds of jobs—something for almost everyone! Here are some of the jobs that computers do well:

- Using disks or tapes, computers can *store* large amounts of information in a very small space.
- Computers can *search* quickly through thousands of pieces of information to find a particular name, word, or number.
- Computers can *sort* information into categories—such as "elephant jokes" and "radish jokes"—or put information into alphabetical or numerical order.
- Long or repetitive *calculations* can be done quickly by a computer.
- Using facts about the past, computers can help people make *predictions* about the future.
- With a printer, a computer can *print* out reports, letters, receipts, and other forms. Best of all, you can make changes and corrections very easily, and each letter or form can be individualized with a different name or number or other information.




BAKE SALE
$ $
RECIPES
CAKE
COOKING
COOK BOOK
STORE
SPECIAL
·FLOUR
·SUGAR
·BUTTER
·MILK
BAKE SAL
Bake Sale
FOR THE COMPUTER CLUB
ALE
20¢ each
15¢ each
25¢ each

The school's Computer Club is having a bake sale to raise money for a new computer. Listed are 15 of the jobs the members must do.

Circle the number of each job that a computer could easily help them with.

1. Coming up with the idea to have a bake sale.
2. Forecasting how much money the club would make if the sale lasted two days instead of just one.
3. Keeping a file of all the recipes to be used.
4. Searching for recipes that will use up the club's supply of apples (left over from their last party).
5. Figuring out how much flour, sugar, and butter to buy.
6. Going to the store for supplies.
7. Keeping track of how much money the club spends on supplies.
8. Making posters using fingerpaints.
9. Rolling out dough for sugar cookies.
10. Figuring out what price should be charged for each item.
11. Setting up a table in the school cafeteria.
12. Writing announcements to send to all the teachers in the school.
13. Persuading people to buy.
14. Printing receipts for customers.
15. Figuring out the profits after the bake sale is over.

Review: ACCESS TO COMPUTERS

Read the program below. Then fill in the blanks in the five sentences that follow the program.

```
 10 If you've completed pages 5 through 77 of
    this book then go to 30.
 20 Complete any pages you've missed.
 30 A$ = "K CO"
 40 B$ = "IN"
 50 C$ = "RS!"
 60 D$ = "TH"
 70 E$ = "TE"
 80 F$ = "MPU"
 90 PRINT D$ B$ A$ F$ E$ C$
100 END
```

1. Line 10 gives a choice; it is a ______________________________ statement.

2. "MPU" is a ____.

3. Line ______________________________ tells you to give output.

4. If this program were executed by a computer, the output could appear on the computer's ____________ or be put on paper by the computer's ____________.

5. The program's output is: ______________________________

GLOSSARY

array an orderly list or table of information which gives each piece of information its own special name; for example, *A5*

branch in a program, a choice of two or more things to do

bug an error in a program

cassette tape a magnetic tape such as the ones used for playing music; used to store information for a computer

catalog *n.* a list of all the programs on a disk or cassette; *v.* a command that tells the computer to show a list of all the programs on a disk or cassette

cathode ray tube *see* CRT

central processing unit *see* CPU

code symbols used instead of the letters, numbers, or words in a message

command an instruction that a computer follows immediately; not part of a program

computer a machine that helps people store and work with information

condition the first part of a decision statement, beginning with IF; for example, IF *there's smoke*, THEN there's fire.

CPU central processing unit; the main "thinking" part of a computer

CRT cathode ray tube; a screen that displays numbers, letters, and pictures for the computer; also called *video monitor*

data information

dead end a step in a flowchart or program that does not lead to the end of the program

debug find and correct a mistake in a program

decision statement a statement that gives a choice of things to do; takes the form IF . . . THEN, as in, "IF you're thirsty, THEN get a drink of water."

disk *see* floppy disk

disk drive a machine that reads the information from a floppy disk and sends it to the computer

END the last statement in any program

endless loop a task in a program that is repeated over and over, forever; also called *infinite loop*

equivalent expression different way of saying the same number

execute follow the instructions in a program or statement; RUN a program

floppy disk a flat object made of magnetic material, used to store information for a computer

flowchart a map or diagram of a computer program, showing the main steps

GO TO an instruction used to send the computer to another line in the program; for example, GO TO 30

GOTO an instruction used by some computers; means the same thing as GO TO

graphics pictures made by computers, either on paper or on video monitors; may include drawings, maps, graphs, etc.

infinite loop *see* endless loop

input what you put into a process; in programming, the information the user gives to the computer

keyboard a machine that lets you give input to the computer by typing letters and numbers; looks like a typewriter

LET an instruction that tells a computer the value of a variable; for example, LET Y = 32

line number the number given to a statement in a computer program; appears at the very beginning of the statement; for example, 10 LET X = 12

LIST a command that tells the computer to show you a list of all the program statements in your program

LOAD a command that tells the computer to find a particular program, put the program into its own memory, and get ready to use the program

loop a task that is repeated

monitor *see* video monitor

NEW a command that tells the computer to get ready for a new program and erases the old program you've been using

output the result of a process; in programming, whatever the computer prints out or does as a result of following the instructions in a program

PRINT an instruction that tells the computer to display letters or numbers on its video monitor or to print them on paper

printer a machine attached to a computer that types out letters and numbers on paper

process a series of operations, actions, or changes that turns some raw material into a final product

program a set of instructions telling a computer how to do a particular job

program statement a single instruction in a program, with a line number at the beginning

result the second part of a decision statement, beginning with THEN; for example, IF you don't watch where you're going, THEN *you'll fall down*.

RUN a command that tells a computer to execute a program

SAVE a command that tells a computer to record your program on a disk or cassette for future use

START the first step shown in a flowchart

statement *see* program statement

string any series of letters

string variable a symbol for a string; usually a letter or a letter and number, plus a dollar sign; for example, A$ or A(6)$

symbol a sign that stands for another thing or idea; for example, ¢ means *cent*

synonym a word with the same meaning as another word; for example, *Insane* is a synonym for *crazy*.

value the number or string that a variable stands for is the value of that variable

variable a symbol—usually a letter—that stands for a number or value

video monitor a screen that displays numbers, letters, and pictures for the computer; also called CRT

ANSWER KEY

p. 5

1. F 5. F 9. F
2. T 6. F 10. F
3. T 7. T
4. T 8. T

1 COMPUTERS

p. 6

1. computer 2. person

p. 7

3. computer 5. person
4. computer 6. computer

Answers will vary.

p. 8

1. 12

p. 9

2. Ouch! *or* Watch it! *etc.*
3. Red means "stop."
4. How are you?

p. 10

1. **Input:** 6 + 3 *or* What is 6 + 3?
   Output: 9
2. **Input:** add 10 test scores, divide by 10 **Output:** the average score
3. **Input:** how many worms will attend *and* how many apples will each worm eat
   Output: the number of apples needed
4. **Input:** today's date *and* your birthdate
   Output: the number of days

Something to Think About:

1. adding
2. adding/dividing
3. multiplying
4. subtracting

p. 11

1. *example:* knitting
2. hatching
3. melting 5. adding
4. erasing 6. dividing

p. 13

1. processing 5. input
2. input 6. input
3. output 7. input
4. output 8. input

p. 14

1. make a milkshake
2. 5
3. 3
4. lines 2 and 3
5. 6 Drink *or* Serve at once.

p. 15

1. 8 3. 17
2. line 8 4. 24

Note: The answer will always be 17, no matter what age is used.

p. 17

1. The robot would spill the water on the table.
2. The robot would spill water on itself.
3. line 5
4. line 6
5. lines 1, 3, 5, 6, 7, 8, and 10 (wherever the word *glass* is used)

2 Move your right foot forward.
3 Put your right foot down.
4 Lift your left foot.
5 Move your left foot forward.
6 Put your left foot down.

Something to think about:
Yes. Add:
7 Start again at line 1.

p. 18

1. computer
2. CPU
3. process
4. input
5. keyboard
6. output
7. video monitor; printer
8. program
9. program statement
10. RUN

2 FLOWCHARTS

p. 19

Note: There are many possible answers.
Here are some suggestions:

1. Plant the seed in soil.
   Make sure it's in a sunny place.
   Whenever the soil is dry, water it.
   Pull out any weeds that sprout.
2. Jack up the car.
   Remove the flat tire.
   Put on the spare tire.
   Let the car down gently, and remove the jack.
3. Choose a book.
   Read the book.
   Make some notes and an outline.
   Write the report.

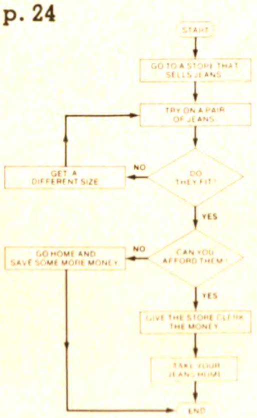
p. 21

1. LOOK UP YOUR FRIEND'S TELEPHONE NUMBER.
2. HANG UP AND TRY AGAIN LATER.
3. WAIT FOR SOMEONE TO ANSWER.
4. HANG UP AND TRY AGAIN LATER.
5. 3

p. 23

1. START 3. 2
2. 3 4. 2
5. (a number from 1 to 9, depending on student's name

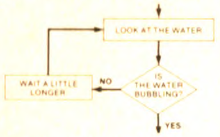
p. 24


START
GO TO A STORE THAT SELLS JEANS
TRY ON A PAIR OF JEANS
DO THEY FIT?
NO
GET A DIFFERENT SIZE
YES
CAN YOU AFFORD THEM?
NO
GO HOME AND SAVE SOME MORE MONEY
YES
GIVE THE STORE CLERK THE MONEY
TAKE YOUR JEANS HOME
END

p. 25

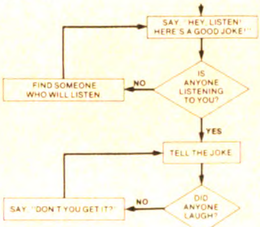
1. an infinite number of times, or as many times as there are sizes of jeans in the shop
2. when you found a pair of jeans that fit
3. TRY ON A PAIR OF JEANS.
   DO THEY FIT?
   GET A DIFFERENT SIZE.
4. It would be possible if the very first pair of jeans fit perfectly.

p. 26


LOOK AT THE WATER
IS THE WATER BUBBLING?
NO
WAIT A LITTLE LONGER
YES

1. LOOK AT THE WATER.
   IS THE WATER BUBBLING?
   WAIT A LITTLE LONGER.
2. when you saw the water bubbling
3. if the water were bubbling the first time you looked

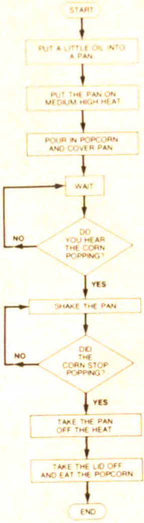
p. 27


SAY, "HEY, LISTEN! HERE'S A GOOD JOKE!"
IS ANYONE LISTENING TO YOU?
NO
FIND SOMEONE WHO WILL LISTEN
YES
TELL THE JOKE
DID ANYONE LAUGH?
NO
SAY, "DON'T YOU GET IT?"

1. SAY, "HEY, LISTEN! HERE'S A GOOD JOKE!"
   IS ANYONE LISTENING TO YOU?
   FIND SOMEONE WHO WILL LISTEN.
2. when someone was listening to you
3. if someone were listening to you right away
4. TELL THE JOKE.
   DID ANYONE LAUGH?
   SAY, "DON'T YOU GET IT?"

p. 29

Some variation and elaboration is possible, but here's one solution:


START
PUT A LITTLE OIL INTO A PAN
PUT THE PAN ON MEDIUM HIGH HEAT
POUR IN POPCORN AND COVER PAN
WAIT
DO YOU HEAR THE CORN POPPING?
NO
YES
SHAKE THE PAN
DID THE CORN STOP POPPING?
NO
YES
TAKE THE PAN OFF THE HEAT
TAKE THE LID OFF AND EAT THE POPCORN
END

3 PROGRAMS

p. 30

1 Open a carton of milk.
2 Tilt the carton so that milk pours into a glass.
3 Pour until the glass is full.

p. 31

1. student's name
2. translation (from English to Pig Latin)
3. student's name in Pig Latin
4. student's name
5. the name of the movie or TV show in Pig Latin

p. 32

1. The eggs, bread, cookies, and crackers would get crushed.
2. Different answers are possible, providing cans and bottles are at the bottom of the bag, and eggs, bread, cookies, and crackers are toward the top.
   1 Open an empty bag.
   2 Put in cans and bottles.
   3 Put in fruits and vegetables.
   4 Put in bread, cookies, and crackers.
   5 Put in eggs.

p. 33

5 Pour the mixture into the pan.
3 Mix with an eggbeater.
1 Break two eggs into a bowl.
6 Cook and stir for a few minutes, until done.
2 Add a little milk and some salt and pepper.
4 Melt a little butter in a pan.

1. scrambled eggs
2. The milk would get cooked, but the eggs would be raw.
3. *Answers will vary.*

p. 34

1. Get into the car.
2. between lines 1 and 2

p. 35

Other answers are possible, but these leave room for additional lines:

30 Fasten your seat belt.
5 If the car door is locked, unlock it.
17 Make sure your hands and feet are clear of the door.

HOW TO CLEAN YOUR ROOM

10 Make your bed.
20 Put away your clothes.
30 Dust your furniture.
40 Vacuum the floor.
50 Throw away old papers.

Additional Lines:

35 Pick things up off the floor.
5 Clear the junk off your bed.
60 Empty the wastebasket.

New line will vary.

p. 36

1. 5, 10, 20, 50, 100, 120, 200
2. 15 or 30-40 Check the water temperature.
   130-190 Dry yourself off.
   110 Rinse yourself off.
   2 or 3; or 30-40 Take off your clothes.
3. All the line numbers listed, in order

p. 37

10 Go the the pizza parlor.
20 (or 30) Decide whether you want a small or large pizza.
30 (or 20) Argue about whether to have anchovies on your pizza.
40 Order your pizza.
50 Wait 10 minutes longer than you were told.
60 When the pizza is ready, pay for it.
70 Take the pizza to your table.
80 Eat the pizza.

p. 38

1. 6
2. 20, 40, and 50
3. 20 Get some dog food.
   40 Put the dish on the floor.
   50 Make sure the dog has a bowl of water to drink.
4. 6

p. 39

10 Open an empty suitcase.
20 Decide what you need for your trip.
30 Put your shoes in the suitcase first.
40 Carefully fold your clothes.
50 Lay them neatly in your suitcase.
60 Close the suitcase.

p. 41

1. wrong order
2. Take out line 40 and make it line 15 instead.
3. wrong instruction
4. 20 Turn off the television and radio.
5. unclear instruction
6. 30 Press the button underneath the candy you chose.

p. 42

1. Dragon's Cave
2. Gold Mine
3. Haunted House
4. right, left, right
5. left, right, right
6. right, right, right

p. 43

1. 2
2. 4
3. white sport socks
4. navy-blue socks
5. START; GO TO THE STORE; DOES YOUR FATHER LIKE SPORTS? (You must buy a pair of socks of some kind.); END

p. 44

1. either eat your dinner or secretly slip it to the robot
2. 20 and 30
3. Eat your dinner.
4. Nothing; *or* go straight on to line 30.
5. you don't like liver and spinach
6. secretly slip your dinner to the robot
7. it is raining
8. put up your umbrella

p. 45

1. it is dark and it is clear,
   we look at the stars
2. it is dark; it is clear
3. yes
4. no
5. no
6. it is cold or it is rainy,
   we stay home.
7. it is cold; it is rainy
8. yes
9. yes
10. no

p. 46

1. yes
2. yes
3. yes
4. no
5. students 10 or older: yes
   students under 10: no
6. yes
7. no
8. yes

p. 47

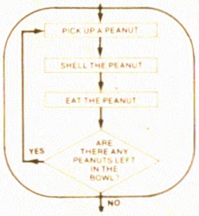
1. you want cola
2. 70; skip lines 40, 50 and 60
3. 10, 20, 30, 70, 80, 90
4. 40; 90; skip lines 70 and 80
5. 10, 20, 30, 40, 50, 60, 90

p. 48

line 10: GOTO 60
line 20: GOTO 90
Line 80: GOTO 110

1. 10, 30, 40, 50, 60, 70, 80,
2. 110, 120

p. 49



Circle lines 20 through 50 in the program.

1. line 50
2. GOTO line 20 and pick up another peanut.
3. GOTO line 60 and throw away the shells.
4. as many times as there are peanuts in the bowl; or, until the bowl is empty
5. lines 10, 60, and 70

p. 50

1. 20, 30, 40
2. an endless number; forever
3. 35
4. you are not tired

p. 51

5. 40, 50, 60
6. 55 IF you have read the last page of the book, THEN GOTO 70.
7. 60 IF you have not read the last page of the book, THEN GOTO 40.
8. the second way (rewriting line 60)

p. 52

1. your name is Lily
2. GOTO 30
3. 10 IF your name is Lily, THEN GOTO 30.
4. branch

p. 53

5. do exercises
6. 10 IF you feel tired THEN GOTO 40.
7. forever
8. no
9. 30 IF you aren't dizzy, THEN GOTO 20.
10. 25

Something to think about: changing line 30

p. 54

1. decision statement
2. branch
3. condition
4. result
5. loop

p. 55

10 Drive out of your garage.
20 IF you want to go to the drive-in movie, THEN GOTO 60.
30 Turn right.
40 Turn left.
50 GOTO 80
60 Turn left.
70 Turn right.
80 Pay the attendant.
90 Drive in.
100 END

4 CODES

p. 56

1. < >
2. < >
3. =
4. < >
5. =
6. 2 × 5; 30 – 20
7. 12 months; 52 weeks
8. infrequently; rarely; hardly ever
9. 10 – 10; nothing; zilch

Something to think about: an infinite number

p. 57

10. 27 < 28
11. 29 > 24
12. 7 > 5
13. 50 < 55
14. 72 < 80
15. 28 = 28
16. 48 < 50

p. 58

1. c
2. d
3. b
4. a
5. Meet me at ten.
6. Destroy this note.
7. 8-9
8. 3-15-13-5 20-15 1 16-1-18-20-25
9. student's name in code

Something to think about: by using letters

p. 59

1. 3
2. 2
3. 7
4. 1
5. 5
6. 10
7. 1
8. 20
9. 9
10. 13
11. current year
12. student's year of birth
13. last answer plus 10
14. 1999

p. 60

1. ANYWHERE HE WANTS TO
2. A GARBAGE TRUCK
3. TO HOLD HIS PANTS UP
4. SO HE COULD HIDE IN THE JELLY BEANS

p. 61

1. nothing
2. WHY DID THE ELEPHANT WEAR RED SOCKS?
3. BECAUSE HER GREEN ONES WERE IN THE WASH
4. BECAUSE D$ B$ ONES A$ E$ THE C$
5. so that spaces will appear between words in the sentence

p. 62

1. A$
2. (a) 26 (b) 26

p. 63

3. vegetables
4. beverages
5. baked octopus, mashed turnip, and cherry pie
6. A(2), D(4)
7. 20
8. 25
9. 26
10. unlimited

p. 64

1. student's height, weight, and age
2. (a) 58
   (b) 112
   (c) student's age
3. 496 + student's weight

p. 65

	1	2	3
J	toothpick sharpener	mattress tester	banana farmer
P	ostrich	alligator	ant
V	helicopter	unicycle	skateboard

The banana farmer is person 3, the ant owner and skateboard rider.
The unicycle rider is person 2, the alligator owner and mattress tester.

p. 66

1. to 3. depend on number of breaths per minute
4. Number of breaths in one day. *(Number will vary.)*
5. H
6. D

p. 67

7. *Numbers will vary.*
8. *Number will vary*; person 3; minute
9. *Number will vary*; person 2; hour
10. *Number will vary*; person 4; day

p. 68

1. friend's name
2. the year the friend was born
3. YOU'RE (friend's age) HAPPY BIRTHDAY (friend's name) FROM (student's name)

5 ACCESS TO COMPUTERS

p. 69

1. program statement
2. command
3. program statement
4. program statement
5. command
6. program statement
7. command
8. command
9. program statement
10. program statement

p. 70

20 keyboard
30 NEW
40 program statement
70 END
80 RUN
90 END

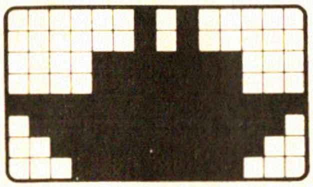
p. 71

1. b
2. a
3. d
4. c
5. e

p. 72

3,2 6,2
4,4 5,4
2,5 7,5
3,6 4,6 5,6 6,6

p. 73



p. 75

2, 3, 4, 5, 7, 10, 12, 14, and 15 should be circled

p. 76

1. decision or branching
2. string
3. 90
4. CRT (or video monitor or screen); printer
5. THINK COMPUTERS!

PAID

SCHOLASTIC INC.

33055

0 78073 02500

ISBN 0-590-33055-1

4.95

0-590-33055-1